Internet Resource Discovery at the University of Colorado

Michael F. Schwartz

Department of Computer Science
University of Colorado
Boulder, Colorado 80309-0430
(303) 492-3902
schwartz@cs.colorado.edu

October, 1992

To appear, *IEEE Computer Magazine*

Abstract

Rapidly increasing global Internet connectivity offers tremendous opportunities for collaboration and information sharing. An important problem in this environment is how to discover resources of interest, such as documents, network services, and people. In this paper we discuss a number of aspects of the resource discovery problem, and summarize results from efforts to address these problems carried out in the Networked Resource Discovery Project at the University of Colorado.

CR Categories and Subject Descriptors: C.2.4 [**Computer-Communication Networks**]: Distributed Systems – *distributed applications*; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *indexing methods*; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *retrieval models, search process, selection process*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software – *information networks*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services – *data bank sharing*; H.3.6 [**Information Storage and Retrieval**]: Library Automation – *large text archives*; H.4.3 [**Information Systems Applications**]: Communications Applications – *bulletin boards, electronic mail*; K.4.1 [**Computers And Society**]: Public Policy Issues – *privacy*.

General Terms: Design, Experimentation, Measurement

Additional Key Words and Phrases: resource discovery, directory service, global distribution, administrative decentralization.

Introduction

The Internet is a collection of several thousand networks interconnecting millions of users at academic, industrial, and government institutions worldwide [1].[1] As such, it offers tremendous potential for collaboration and sharing of resources, such as documents, software, data, and network services. In this environment, an increasingly important problem is how users can discover what resources are available. This *resource discovery* problem is important, because it is an enabling aspect of wide area collaboration. Without the ability to discover resources, users perceive only a very limited fraction of the full potential for sharing resources and collaborating with colleagues.

An environment as large as the Internet places stringent scalability requirements on the algorithms used to locate and track resources. A common solution to this problem is to organize resource information into a hierarchy. Unfortunately, the organization of a hierarchy becomes convoluted as an increasingly wide variety of resources is registered, and requires users to understand how the increasingly deeply nested components are arranged. Moreover, hierarchical information is only efficiently searched according to its primary organizational attributes. For example, in a hierarchy organized by country and institution at the top two levels, searching for software packages according to functionality would be infeasible, because doing so would require contacting every server in the system. Such global operations would be prohibitively expensive in a system spanning hundreds or thousands of servers worldwide.

The administrative decentralization of the Internet also means that resource information may originate from many different sources, in a variety of formats. While standards are helpful in this regard, it is difficult to specify standards that are both globally adopted and technologically current. As an increasingly diverse collection of institutions contribute to the global "information infrastructure," it must be possible to accommodate multiple ways of formatting and organizing information.

Another consequence of the decentralization of the Internet is that in many cases resources are not formally advertised. For example, many Internet hosts provide open access to a variety of software and documents, but administrators on these hosts usually do not register this information in a directory service. Typically, such resources are announced in one-time-only, textual electronic mail or news messages. Even if resources are formally registered, some resources are of value for only a short time, perhaps a few weeks. Traditional systems for organizing information (as used by librarians, for example) are not well suited to such rapidly changing information.

Beyond the issues of scale, decentralization, and rapid change discussed above, another issue that can complicate resource discovery is the existence of a commercial environment. In such an environment, the resource discovery mechanisms must provide fair access among competing information providers. This issue will become increasingly important in the next few years, as the U.S. Regional Bell Operating Companies enter the information services market, and the Internet begins explicitly allowing commercial traffic.

In this paper we discuss a number of different aspects of the resource discovery problem, and summarize efforts to address these problems in the context of the Networked Resource Discovery Project at the University of Colorado. We begin with a survey of a number of well-known Internet resource discovery systems.

Internet Resource Discovery Systems

A number of Internet sites run centralized servers that support queries about people and other information across the Internet. One prominent example is the WHOIS service, used by Network Information Centers (NICs) and other organizations to maintain databases of registered users and network sites [2]. Because each WHOIS server collects geographically distributed information into a single database, it provides a good focal point for registrations and searches. However, any one server contains only the small fraction of Internet users and sites that have registered with that NIC, and the information gets out of date because people often forget to

[1] Throughout this paper we use "**I**nternet" to refer specifically to this collection of networks. We use "**i**nternet" to refer to general internetworks.

inform the NIC when their information changes. Moreover, because each WHOIS server is run independently of the other WHOIS servers (without coordinating content or format), users must explicitly deal with the distribution and inconsistencies between servers. Finally, as the Internet continues to grow, a centralized directory will become a bottleneck and critical point of failure.

The Consultative Committee on International Telephony and Telegraphy (CCITT) and the International Organization for Standardization (ISO) have jointly developed a distributed directory service standard called X.500, which describes a hierarchical name space, with provisions for caching, authentication, and replication [3]. Each participating site maintains directory information about resources at that site, as well as administrative information needed for traversing the tree and maintaining proper distributed operation. There are a number of implementations of X.500 available, and field trials are underway to demonstrate interoperability between the implementations. While X.500 is defined as part of the ISO protocol suite, it can run on top of the Internet through an implementation of the ISO transport service on top of the Internet Transmission Control Protocol (TCP).

A disadvantage of X.500 is that it requires a non-trivial level of effort for a site to install the server software and populate the database with useful information. An increasingly popular way to overcome such problems is to build systems that provide directory service based on existing sources of information, without requiring effort from individual site administrators. This technique is the basis of the archie service, which maintains a list of approximately 1,100 "anonymous" FTP[2] archives worldwide, and builds a database of retrievable files by performing recursive directory listings at each site once per month [4]. These sites currently contain about 150 gigabytes of information, in approximately 2.6 million files. Users can query this database via several interfaces from any of 13 replicated archie servers worldwide, using regular expressions and other types of queries.

Because archie provides an index, searches are not constrained by the hierarchical nature of Internet host names. Users simply specify regular expressions describing the names of files they are trying to locate. In contrast, there is no way for a user to search the X.500 directory service with a similar *flat* global search. The user must browse the X.500 tree to locate information. Some X.500 implementations support flat searches through parts of the tree, but not global searches.

While archie allows users to search for files, the Prospero file system allows users to organize files according to their personal preferences, by creating *views* [5]. For example, a user might create a view concerning a particular research topic, and populate that view with links to relevant files distributed around the Internet. Other users can then browse this information. The World Wide Web (WWW) system also allows users to organize and access information without concern for its distribution [6]. However, WWW supports two separate discovery models. Part of the information space is based on a hypertext paradigm, where users can explore information by selecting hypertext links to other information. Other parts of the information space consist of indices, which the user encounters while exploring the hypertext space. The user accesses such indices using a flat search paradigm.

The Wide Area Information Servers (WAIS) system allows users to deploy, search, and retrieve documents and many other types of information from indexed databases throughout the Internet [7]. While the archie index contains only file names, WAIS indices contain keywords for every word that appears in textual documents (other than common words like "the"). WAIS divides its indices among the servers that provide information, rather than using one global index. This solution has better scalability properties than archie's single index, but it also means that users cannot use flat global searches. Instead, they must first search an index of servers, and then select a few servers to search.

The Internet Gopher provides a uniform user interface to many different types of network information [8]. Users are presented with a hierarchical menu system, allowing them to access information from many of the systems listed above, plus various online telephone books, library catalogs, and other data.

The Corporation for National Research Initiatives introduced the notion of a Knowbot™ (Knowledge Robot), which can launch searches for information in a network, possibly replicating itself onto other nodes. Droms implemented an Internet user directory service called the Knowbot Information Service [9], which understands the input and output formats of a number of directory services (such as X.500 and WHOIS), and translates user

[2] FTP is the Internet standard File Transfer Protocol. Anonymous FTP is a convention for allowing Internet users to transfer files to and from machines on which they do not have accounts, for example to support distribution of free software and technical reports.

requests as needed to access these services.

Name-Based vs. Attribute-Based Searches

If a user knows the name of a resource, a name-based or *white pages* query can be made. If the user knows only some attributes of the resource (e.g., a description of a document's topic), an attribute-based or *yellow pages* query is required. Attribute-based searches are more difficult to support in general, because there are many different ways that users may describe resources in search requests.

Internet resource discovery systems often support some combination of white pages and yellow pages queries. For example, archie is primarily a white pages service, because it indexes file names. However, often the directory names in a tree contain some attribute information (for example, source code for X window system applications may reside under a directory called "Window-Systems"). Because of this, some yellow pages searches are supported. Nonetheless, such searches tend to be more haphazard than searches that specify the name of a file being sought.

Below we discuss two Networked Resource Discovery Project systems. While each system supports a combination of white pages and yellow pages searches, the first system is primarily a white pages service, and the second system is primarily a yellow pages service.

Internet User White Pages

We have built and experimented extensively with a tool called *Netfind*, which locates electronic mail addresses and other information about Internet users, based on a number of widely distributed sources of simply structured information [10]. Using distributed information avoids difficult problems of consistency and transfer of authority that are inherent in mechanisms that rely on building centralized databases to hold the information. The ability to use simply structured information is important in an environment where global agreement about structured information formats is difficult to achieve.

Figure 1 illustrates the Netfind search algorithm. The example search specifies the name of the current paper's author ("schwartz") and a set of keywords describing the institution ("colorado university"). These keywords are used to access a *seed database*, to obtain hints of potential *domains* to search (such as departments within a university or company). This database is gathered by continually monitoring and cross-correlating a number of data sources, including USENET electronic bulletin board messages, UUCP network maps, WHOIS data from several Network Information Centers, recursive listings of the Domain Naming System (DNS), logs from various network services, and information supplied by users. In the figure, a number of domains match the search (indicated by the ellipsis). The user is asked to select at most three domains to search (not shown in the figure), after which point Netfind searches each domain as follows. Netfind first looks up the domain in the DNS, to locate name servers for the domain. These servers often run on central administrative machines, with accounts and mail forwarding information for many users at a site. Netfind then queries the Simple Mail Transfer Protocol servers on the machines where the DNS servers run, in an attempt to find mail forwarding information about the specified user. If such information is found, the machines to which mail is forwarded are queried using the *finger* service. Search operations proceed in parallel, to increase resilience to host and network failures.

Netfind can often find a user even if the remote site does not support all of these services, or if some steps in the sequence fail. For example, if the finger service is not supported, mail forwarding information may sometimes still be found. Or, if no mail forwarding information is found, Netfind attempts to finger some of the machines listed for that domain in the seed database. Similarly, Netfind can proceed without information about name servers. This ability to function in the presence of failures or partial remote service support provides a degree of fault tolerance without requiring global agreement about what services are supported at each site. This technique allows Netfind to locate information about more people than other Internet user directory services (see sidebar). Moreover, because many different institutional keywords will lead to the same seed database records and Domain information, it is usually quite easy to "guess" keywords that will succeed for any particular search.

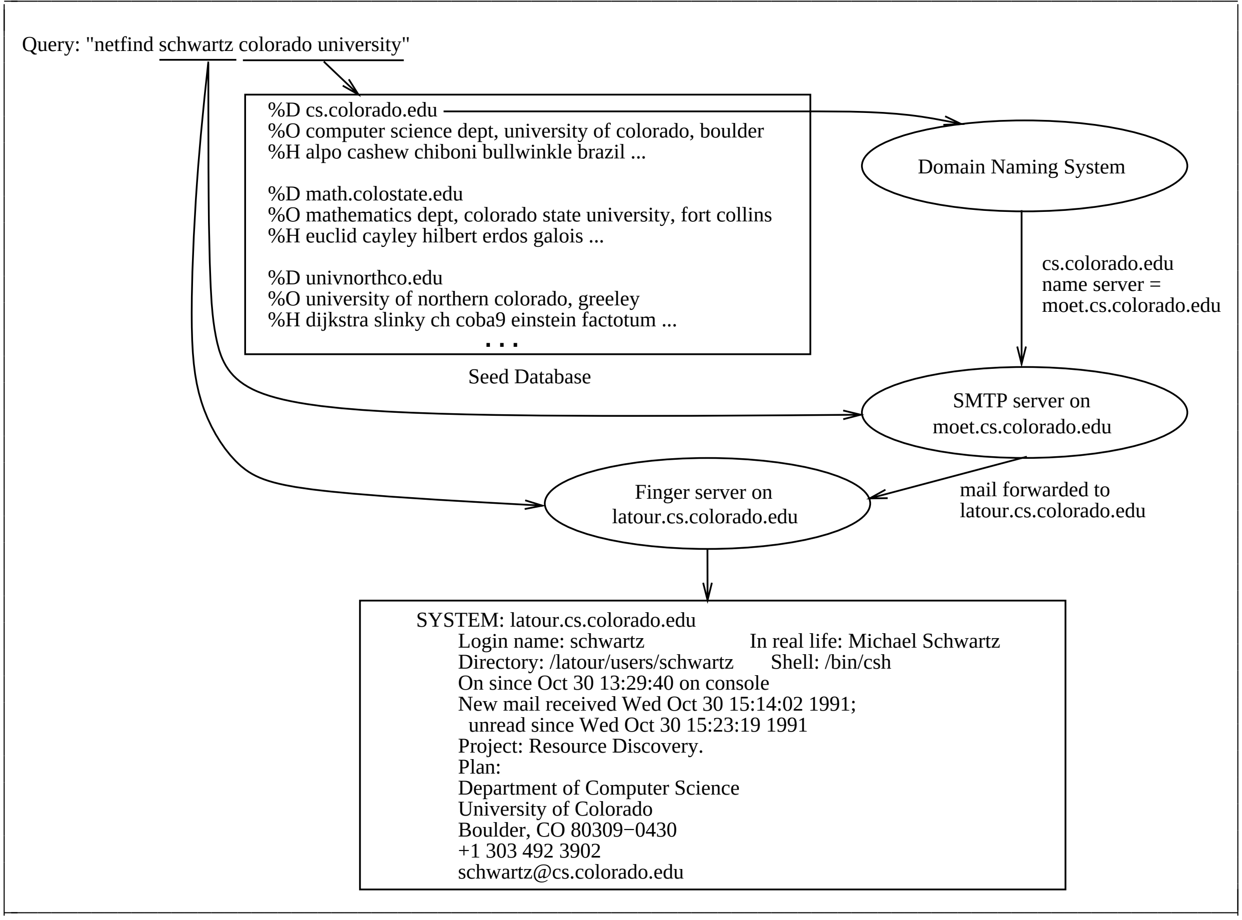

Figure 1: Netfind Search Algorithm

Internet File Archive Yellow Pages

Archie supports Internet FTP file discovery using automatically gathered directory information. While this technique provides an immediate base of useful resource information, it primarily supports white pages searches. Effectively supporting yellow pages searches requires a fairly large and carefully chosen set of attributes to be associated with any resource, to match various keywords that users may use. Because of the labor intensity of this requirement, we explored an approach to Internet file archive discovery that allows incremental organization of the resources, based on the efforts of a geographically distributed collection of people, and a range of different information sources of varying degrees of quality [11].

In our prototype system, four levels of information quality are supported. At the lowest level, the system monitors USENET electronic bulletin board articles using some simple heuristics to recognize announcements about public archive sites. This information provides a simple keyword-based index of files throughout the Internet. The next higher level of information quality is provided by per-user and per-user-site caches, which record resources that have been found by individual users during their explorations. At the next higher level, resources are described using archive-site-resident databases, with individual resources described according to their conceptual roles. An example of the contents of one such database is shown in Figure 2. The fields in each record contain file names, path names, the FTP command needed to cause the file transfer, a description of the resource, and keywords describing the resource. These keywords and descriptions can support effective yellow pages searches. This database is constructed with the aid of automated tools that generate some basic keywords from file and directory names, which can be modified manually to improve keys.

At the highest level of information quality, users who share common interests can build views that superimpose organization on the resource space according to their particular preferences, similar to Prospero views. Unlike Prospero, our system incorporates user-provided attribute specifications into each view, and supports flat

(Sidebar) Measurements of Netfind

Netfind's tolerance of partial remote service support allows it to locate information about 5.5 million people in 9,258 domains worldwide, broken down as indicated in Table 1. These numbers are significantly larger than those for services (such as WHOIS or X.500) that rely on explicit registrations. Moreover, because the seed database contains information about many sites that are not currently connected to the Internet, Netfind can often locate users at sites immediately after they connect to the Internet. Indeed, the seed database contains nearly three times as many domains in nearly twice as many countries as those counted in Table 1.

Top-Level Domain Name	Description	Searchable Sub-Domains	Top-Level Domain Name	Description	Searchable Sub-Domains
edu	U.S. Educational	3,340	za	South Africa	25
com	U.S. Commercial	968	pl	Poland	24
de	Germany	836	mx	Mexico	24
ca	Canada	603	kr	Korea	21
au	Australia	378	ie	Ireland	20
uk	United Kingdom	360	gr	Greece	20
jp	Japan	350	is	Iceland	18
se	Sweden	263	pt	Portugal	13
gov	U.S. Government	248	hk	Hong Kong	13
mil	U.S. Military	191	cs	Czech & Slovak Federated Republic	13
fr	France	184	us	United States	11
no	Norway	181	cl	Chile	9
nl	Netherlands	168	ee	Estonia	7
net	Named by Network	163	sg	Singapore	6
it	Italy	129	hu	Hungary	4
org	Non-profit	122	pr	Puerto Rico	2
fi	Finland	85	lu	Luxembourg	2
dk	Denmark	80	yu	Yugoslavia	1
at	Austria	68	ve	Venezuela	1
ch	Switzerland	63	tn	Tunisia	1
es	Spain	58	th	Thailand	1
br	Brazil	43	int	International	1
nz	New Zealand	42	arpa	Old ARPANET names	1
tw	Taiwan	36	ar	Argentina	1
be	Belgium	33	aq	Antarctica	1
il	Israel	26			

Table 1: Counts of Domains Searchable by Netfind

Most user directory services depend on auxiliary databases, which are difficult to keep up-to-date. Because Netfind seaches the machines on which users do their daily computing, it typically (though not always) locates very timely information. Often, Netfind locates users on their "home" workstations, where they have logged in within the past few days.

A potential downside of Netfind's approach is increased search costs. To gauge these costs, we built a mechanism into Netfind so that after each use, it would transmit network load and other measurements to a logging server at the University of Colorado. We distributed this version of Netfind to researchers at 24 institutions, and recorded data for six months. We found that the mean network load per search was 136.33 packets. Combining this with usage frequency and Internet size measurements, we estimate that widespread use of Netfind will contribute less than 0.5 percent to Internet load.

Netfind is in active use by people in 3,272 domains in 37 countries. Internet users can access a public Netfind server by telnet to bruno.cs.colorado.edu, and logging in as "netfind" with no password.

searches using an automatically constructed index of each view.

The problem remains that a user must discover appropriate views to search when trying to find a resource. To address this problem, an area of future research involves experimenting with a means of automatically

interelating views, based on the interest clustering algorithm developed in connection with our electronic mail

```
%A /pub/X/contrib
%B andrew
%C get andrew
%D Andrew -- X windows interface prototyping tool, from Carnegie Mellon University.
%K user interface source code cmu tool prototype X windows browse
```

Figure 2: Example Internet File Archive Database Contents

users to search for resources without having to know what views exist.

Class Discovery vs. Instance Location

Most resource discovery systems do not distinguish between discovering some *class* of resource (such as a software package with specified functionality) and locating an appropriate *instance* of a resource (such as a file containing that software on a particular Internet host). Ignoring this distinction can burden users with extraneous information, and waste network bandwidth. For example, in response to a search for a particular file, archie provides the user with a list of copies from all over the world. Often, the list is longer than a user wishes to browse. Even if the list only contains a few entries, users must attempt to choose a good site from which to retrieve the file. People typically choose a site in their country, or a site they have found to have good response time during previous file transfers. As the Internet becomes increasingly complex, making this determination manually will become difficult. Optimizing for network bandwidth, reliability, cost, and other factors will need support from the network layers responsible for routing, flow control, and accounting.

To some extent, this problem can be reduced by providing popular files on large archive servers at focal points in the network, such as regional Network Information Centers. For example, one Australian archive site uses Prospero to provide pointers to files on many large FTP archives around the world. This supports an illusion that the archive contains copies of the files on all of these archive sites, without requiring that the files be stored locally. Once a file has been retrieved, however, it is cached on the server. The expectation is that users in Australia will use this site for retrieving files, and save bandwidth on the Australian long-haul links for popular files. Unfortunately, if people outside of Australia use this archive, a file not already in the cache can be transferred across the link twice.

We are exploring a different approach, focusing on the questions of where to place file replicas, and which replica to chose during retrievals. The basic idea, suggested by Phil Karn of Qualcomm, Inc., is to distribute files in response to requests for them, caching them at intermediate nodes along a dynamically developed spanning tree of the Internet. Doing so can potentially reduce Internet traffic substantially.

A number of issues must be addressed in developing such a mechanism, including cache location, cache replacement policy, accountability of cache contents (to protect against accidental or malicious replica modification) and cache consistency. As a first step, we measured Internet file transfer traffic, to gauge the extent to which duplicate network traffic flows across various Internet gateways. Our preliminary results indicate that a large proportion of file transfers are caused by retransmissions of a small number of very popular files (see sidebar). We are currently carrying out a more involved version of this study, in which data we collect from the NSFNET backbone will be used as trace inputs to simulations of distributed caching algorithms.

(Sidebar) How Many Times Do Popular Files Get Transferred Across Internet Links?

From December 6-19, 1991 we measured file transfers between the University of Colorado and the rest of the Internet, to determine the extent of duplicate transmissions [12]. The measurement software sampled 20 bytes from each FTP file transfer, as a probabilistic test of file identity. (Equality cannot be established by file names, because they are not unique across Internet nodes.) During the monitoring period, 39,324 transfers of 23,622 different files were observed. Of these files, 21.10% were transferred more than once, with the result that 39.93% of the transfers were for files that had previously been transferred. Looking at transfers by file size rather than count, 54.37% of the 8,321,000,202 total bytes were transmissions of files that had previously been transferred across the gateway we monitored. Combining this measurement with the observation that FTP currently accounts for 45% of the bytes transmitted on the NSFNET backbone, a topology-based distribution and caching mechanism could reduce Internet traffic by as much as 25%.

We found that files transferred multiple times were significantly larger on average than other files. Also, duplicate file transfers consisted of 21.10% of the files that were transferred, with only 6.30% of all files generating 19.30% of the file transfers. Because a small number of large files were transferred a large number of times, a topology-based distribution and caching mechanism could reduce Internet traffic.

We also found that 8.21% of duplicate file transmissions were caused by user errors when transferring binary data, because of the default ASCII conversion mechanism of FTP. Moreover, only 32.16% of files that were transferred were compressed. If FTP were modified to automatically compress files, a reduction of 4.00% in total network traffic would result. While this compression would be worthwhile, our above figures indicate that caching could potentially reduce traffic much more substantially. Nonetheless, the observations about incorrect binary data transfers and lack of compression indicate that Internet data dissemination could be made more effective if support for data conversion and compression were integrated into applications in a fashion that hides the details from users.

A final interesting statistic is that only 44% of FTP connections resulted in file transfers. Most other connections appear to have been directory requests. Whether caused by manual requests or archie automated directory retrievals, this statistic underscores the need for resource discovery support in the Internet.

Global State Discovery

Rather than searching for a specific resource, sometimes what is desired is to discover part of the state of a large system. For example, a system administrator may wish to discover the set of all file servers whose queue lengths exceed some value. Using this type of formulation, a number of problems of network and system management can be cast as resource discovery problems.

Our Fremont[3] project focuses on this aspect of resource discovery, allowing a user to discover information about network characteristics such as topology, congestion, routing, and protocol usage [13]. Like Netfind, Fremont uses a number of protocols and information sources, to support discovery in the absence of global agreement on any one protocol or information source. However, Fremont uses a much more extensive collection of protocols and information sources. An explicit component of our approach is the recognition that different sources of network information have different characteristics with respect to timeliness of discovered information, discovery expense, danger of causing network problems, and completeness of discovered information. In contrast, network management systems based on a single standard are limited to the characteristics provided by that standard. The Simple Network Management Protocol, for instance, takes the perspective that a network is a collection of instrumented devices from which one can measure such quantities as packet flow rates, routing table contents, etc. A different set of characteristics may be detected if passive packet monitoring servers are installed in the network, or if the network management system supports directed probes into a

[3] John C. Fremont was a versatile explorer in Colorado and other parts of the western United States in the 1800’s.

network. Moreover, discovering information from multiple sources can allow more information to be collected than may be available from a single protocol. Finally, inconsistencies in the information discovered from different sources can help uncover network problems.

The Fremont architecture uses a collection of servers distributed around portions of the internet being instrumented. Each server periodically executes a set of discovery protocols to maintain information about the network segments for which it is responsible. A discovery protocol scheduler allows the system administrator to specify which discovery protocol modules will be scheduled and how frequently, according to individual preferences about the importance of timeliness of discovered information, expense of discovery, etc.

To support scalable operation, the Fremont architecture provides for caching of discovered network information. In addition, queries may specify predicates, to avoid retrieving unchanged or unneeded information. For example, a query may request information about all of the hosts on a particular network segment that have been discovered since the most recent cache entry. In addition to enhancing scalability, this mechanism can improve the responsiveness of browsing operations, for the case where users are browsing parts of an internet that have been viewed recently.

Our prototype system currently runs on the University of Colorado local area internet. We have implemented mechanisms to discover connected hosts and host types, network routing information, and network transmission failures, using several different protocols for discovering each of these pieces of information. We plan to broaden the scope of the prototype, to support discovery of portions of the Internet.

Resource Characterization and Information Distribution

Two basic problems underlie the resource discovery systems discussed so far: characterizing the resources of interest using name/attribute descriptions, and distributing this information so that it can be searched flexibly and efficiently. The approach taken by the Networked Resource Discovery Project for these problems is based on what we call the *Two Phase Discovery* paradigm, in which both characterization and distribution/search are discovery processes. That is, each phase has a dynamic, exploratory nature. Rather than depending on manually registered data about resources, we use mechanisms that characterize resources by extracting data from the resources themselves, or from other existing sources of information. Rather than using a hierarchy or other relatively static organizational structure to determine how information is distributed and searched, we use information where it naturally resides. (The information may naturally reside in a hierarchy. However, we do not limit users to searching according to the organization of this hierarchy.) Moreover, because automatically extracted information can contain errors, we improve data quality using *data mining* techniques, which consider the context, semantics, and redundancy of the data.

WAIS and archie exhibit some aspects of the Two Phase Discovery paradigm, since they extract information automatically from resources. WAIS also improves the information by eliminating common words and generating relevance weightings by frequency of occurrence. However, because WAIS operates on very general information (textual documents about any topic), it only exploits redundancy characteristics of human language text. Given a more focused resource discovery problem, more sophisticated characterization is possible. For example, Netfind improves the quality of its seed database by comparing the organizational descriptions discovered from different data sources for each domain, and selecting descriptions based on frequency of occurrence as well as semantic information about the domain names (e.g., by examining departmental abbreviations like "math.", and country codes like ".jp"). Moreover, Netfind exploits relationships between the seed data and sources of data consulted at the sites where searches are performed (the Domain Naming System and the Simple Mail Transfer Protocol), to narrow the scope of each search to a few promising machines. Similarly, our Fremont system, our Internet file archive discovery system, and our Essence file indexing system [14] each exploit semantics, redundancy, and context in the information they discover, to improve information quality.

Below we discuss two Networked Resource Discovery Project efforts that illustrate the Two Phase Discovery paradigm in more depth. The first effort focuses on dynamic resource characterization, while the second effort focuses on dynamic information distribution/search.

Dynamic Resource Characterization: Discovering Shared Interests Among People

The most common approach to resource characterization is to register resource descriptions manually. This approach is used, for example, when loading an X.500 Directory System Agent. (A DSA might be loaded automatically from another database, but the original database will almost invariably have been created manually.) Manual registration is also used by Prospero, where users create views of existing files to organize them into related collections. Manual characterization provides good control over what data is registered for each resource. This may be important for controlling what data is visible, or for providing highly conceptual descriptions. On the other hand, manual characterization is painstaking and error-prone in a large, dynamically changing environment like the Internet, and the information produced can quickly become dated and incomplete.

We explored an approach to dynamic resource characterization in a project to support discovery of people with particular interests or expertise. In this case, manual characterization would correspond to building interest group lists from explicitly registered data. Unfortunately, doing so assumes one knows what lists should be built, and who should be included in each list. Therefore, we explored an approach that instead deduces interests from the history of electronic mail communication, using a set of heuristic graph algorithms [15]. Using this approach, a user could search for people by requesting a list of people whose interests are similar to several people known to have the interest in question. This technique can support a fine grained, dynamic means of locating people with related interests. The set of possible interests can be arbitrarily specialized, and the people located will be appropriate at the time of the search, rather than at some earlier time when a list was compiled.

To explore this approach, we collected electronic mail "To:/From:" logs from 15 sites around the U.S. and Western Europe for two months, and experimented with a variety of algorithms to analyze the data. (The mail messages themselves were not monitored.) The basic data analysis methodology involved applying a function to each node in a graph, to compute an *interest distance* from a particular starting node. This function provides a ranking of nodes by closeness of interests to the starting node. Before applying this algorithm, we used another algorithm to derive a subgraph that eliminated "noise" caused by the sampled nature of the original graph. By applying various interest distance functions starting from nodes we knew personally, we eventually selected a function that could isolate many people who shared interests with the starting node.

As a concrete example, Table 2 shows the closest 12 entries computed around the author. As can be seen, in most cases the isolated people had interests related to the author's, whose interests lie primarily in networks, distributed systems, and privacy/security. The people in this list were not trivially derivable from the communication graph. Several of the people were not known personally by the author, indicating that the algorithm uncovered relevant people with whom the author had never exchanged electronic mail. In fact, Table 2 includes one person whom the author did not know at the time of this study, to whom the author was later introduced by a third party, because their work was related. Moreover, the list omits a number of people with whom the author did communicate, whose interests are not closely related to the author's. Finally, many of the people in the list were not located at any of our data collection sites. The algorithm was able to derive information about these people given data about only a tiny proportion of the total electronic mail community (15 sites around the world).

This interest clustering algorithm does not directly indicate how people are related, because people can be related by many different shared interests. However, by starting with several people known to have a particular interest and intersecting the graphs formed by running the algorithm on each person, one can derive a set of people who likely share that particular interest. Experimenting with this idea produced very promising lists of people related to each chosen starting person, concerning shared interests in such closely related areas as distributed computing, networks, and naming. By specifying only a few "seed" users, many other highly relevant people were located. This idea is similar in spirit to the WAIS notion of relevance feedback for finding data similar to data that has already been located.

Clearly, this resource discovery technique raises significant privacy issues. At the same time, it offers intriguing possibilities for supporting distributed collaboration, by extracting implicit organizational information from a communication graph. We believe there are situations in which the algorithm can be applied without invading privacy. For example, the algorithms could be used to provide an implicit organizational index of shared data in a file system, based on logged accesses to the various documents. Such a mechanism would be acceptable if users were explicitly aware of it when accessing files and could decline its use, just as users of current electronic bulletin board systems are aware that their postings are public information.

Relationship to Schwartz	Distance from Schwartz
Schwartz	0.00
Graduate student where Schwartz attended graduate school, studying networking	0.77
Theory professor who attended graduate school with Schwartz	0.85
Unknown student at a Southwestern U.S. university	0.86
Industrial systems researcher who attended graduate school with Schwartz	0.88
Schwartz's Ph.D. advisor (interested in performance and distributed systems)	0.90
System administrator at an East Coast U.S. university	0.90
Systems and security researcher at a Midwest U.S. university	0.91
Ph.D. advisor of Schwartz's Ph.D. advisor (interested in performance and systems)	0.92
Performance and Systems researcher at a West Coast U.S. university	0.92
System administrator at an East Coast U.S. university	0.92
Head system architect, government research laboratory	0.92

Table 2: Top of Computed Aggregate Specialization Graph Surrounding Schwartz

These algorithms provide an example of the Two Phase Discovery paradigm for characterizing relationships among people. Understanding the semantics of electronic mail names was used to eliminate "noise" caused by the sampled nature of the original graph. Moreover, relationships among colleagues were inferred because of redundancy of neighbor information in the graph. Together these techniques allow characterizing information about people (or other resources) to be inferred automatically.

Dynamic Information Distribution: A Probabilistic Approach

The second half of the Two Phase Discovery paradigm is the information distribution/search problem. As a point of departure, we observe that the most straightforward approach to this problem is to centralize the information. This approach is taken by archie to store anonymous FTP directory listings, and by WAIS to maintain a directory of WAIS servers. While centralized information has worked quite well in these systems to date, a central server can become a performance bottleneck and a critical point of failure as the scale of the system increases. These problems have moved the Internet community to create replica archie servers. Doing so distributes the load, but creates an auxiliary problem of maintaining consistency among replicas.

To reduce these scalability and consistency problems, one can chose a solution where only parts of the resource data are maintained on any particular server. A common approach is to impose some organizational properties on the data, and distribute data according to these properties. For example, X.500 divides information hierarchically. The tree is divided by country at the top level, and by administrative organization (company, university, etc.) at the next level down. Because the information in a hierarchy can be divided into arbitrarily many pieces, hierarchical directories scale well. Yet, as indicated in the Introduction section, it is only efficient to search hierarchical information according to the one way it is organized.

One can mimic the effect of representing multiple search criteria in a hierarchy by maintaining separate structures with symbolic links to the "main" data. For example, the Prospero file system uses symbolic links to provide views of information distributed among anonymous FTP and other Internet file systems. While this technique provides a good environment for browsing information, searches still involve expensive distributed operations.

The basic problem with the approaches discussed so far is that searches are only efficient if the search criteria are represented in the relatively static structures that organize the resource information. To address this problem, we explored a means of disseminating resource information and distributing search effort in such a fashion that information migrates to where it is needed, rather than placing information according to a static organizational structure. Our approach involves the use of probabilistic algorithms for constructing and searching a resource graph [16]. The goal is to locate a few resources that match a user's search request, rather than to support exhaustive searches. In a large network, users usually do not need exhaustive answers. We also assume that it is desirable to return different answers to the same query across search sessions. If consistent responses to queries are desired, one could build a front-end that caches results.

Based on these assumptions, we designed a protocol to support a set of *agents* in organizing and searching the resource space. Agents maintain pointers to sources of resource information, and access these sources via intermediary *brokers*, which enforce access control policies and provide a standard interface to the resource information. While agents are part of the network infrastructure, each broker belongs to the organization whose resource information it exports. This functional breakdown is illustrated in Figure 3.

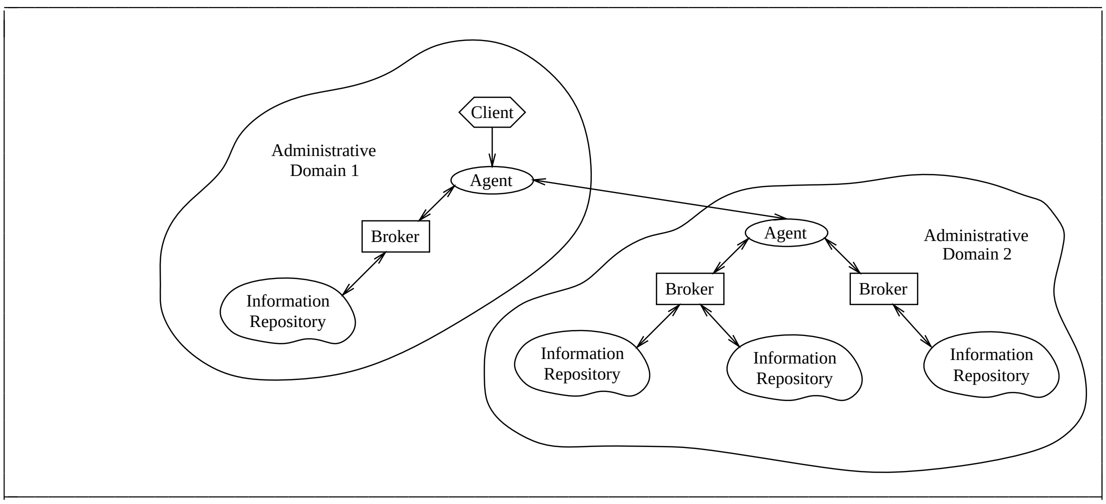

Figure 3: Architecture of Probabilistic Distribution/Search Mechanism

Brokers store whatever information is needed to search and access resource information, such as resource attributes, textual descriptions, and pricing information. Rather than having an administrative body specify how the space is organized, agents organize the space dynamically, according to what resources exist and the types of searches users make. Agents use a probabilistic *Sparse Diffusion Multicast (SDM)* primitive to disseminate information about resources at uniformly distributed, randomly chosen nodes around the network. When an agent receives an SDM advertisement, it exchanges its cache contents with the originating agent (i.e., each updates its cache with information from the other's cache). SDMs are also used to route search requests around the network.

Randomly disseminating resource information is intended to place the information within a reasonably small neighborhood of any agent in the network, so that during searches it is likely that the information can be found using simple random probes. The probabilistic nature of this approach also supports fair access among competing information providers. Because the types of resources that exist and the searches users request are not random in practice, a cache management policy is used to prefer popular resource information. Using this policy, a search initiated at a random agent may cause some random search behavior at the start of the search, until an appropriate agent is reached. At that point, searches proceed in a more directed fashion. If a user continues to use the same agent, over time that agent will maintain pointers to sources of the types of information for which that user often searches.

To analyze the effectiveness of this approach, we built a detailed simulation of the agent protocol. While brokers are an important part of the model, the main focus of this research is on the agent protocols, and hence brokers were abstracted out of the simulation. The simulation computed the proportion of time a resource that had been advertised could be found, as a function of a number of different variables, including the number of agents in the network, the number of different resources, the fan out of sparse diffusion multicasts and the depth of recursion used during searches, cache replacement policy, cache size, and number of cache exchanges before a search was initiated. The results indicate that this approach can support a resource discovery for an environment roughly the size of a country, with several thousand sites participating in resource registration and searches. For example, Figure 4 indicates the differential effectiveness of First In First Out (FIFO) and Least Frequently Used (LFU) cache replacement policies with 5,000 agents, as a function of the number of cache exchanges before a search was initiated. This plot shows that searches succeed a majority of the time with an average of only .2-1 cache exchanges per agent with LFU cache replacement. FIFO cache replacement performs poorly because it has no basis for making good use of an increasing number of cache exchanges.

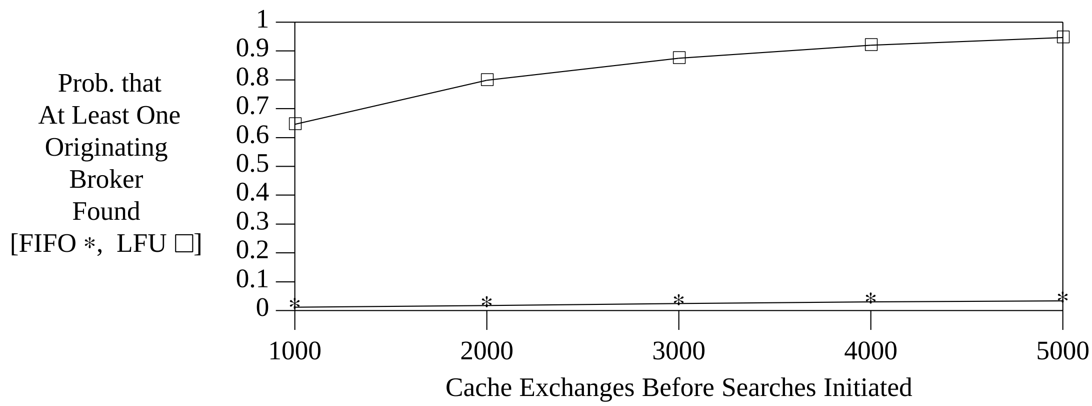

Figure 4: Probabilistic Distribution/Search Effectiveness

Social Issues

Resource discovery raises a number of social issues, particularly concerning privacy. For example, because Netfind searches for users without requiring explicitly registered directories, some people may feel it poses a privacy invasion. We believe it does not, for several reasons. First, Netfind uses publically available information. If a user or site does not want to be located, they can simply disable the network services that Netfind uses (which were, after all, created to allow users to retrieve this information). Second, Netfind only locates addressing information about a person. No personal information is collected, unless a user chooses to make that information available. Finally, Netfind does not provide a way to discover lists of people (such as a list of all engineers at a company). Users can only locate individuals.

In general, any resource discovery system poses potential privacy problems. There have been times, for example, when archie users have discovered and broadly advertised anonymous FTP files that were only intended for limited distribution. This example underscores the fact that people need to be aware of the privacy implications of operating in a large network environment. In this sense, presence on a large network is similar to membership in a society: with the benefits come some loss of individual privacy.

While security mechanisms may be imposed to preserve some degree of privacy in some cases, in many cases such mechanisms are either difficult to provide, or of questionable merit. We believe that privacy is essentially a social issue, and as such requires careful consideration about the policies that will manage the technical solutions. For example, as long as the clustering algorithm we developed in our electronic mail study is only applied when users are explicitly aware of its use, users can chose to decline participation. This system could be used, for example, to build an index of technical documents being shared by a project group.

Beyond privacy, another social issue raised by resource discovery is the assumption (needed for most resource discovery projects) that the Internet will continue to grow and evolve as a medium for supporting wide area collaboration. However, in response to a number of well publicized events over the past few years, many sites have imposed a range of mechanisms to limit their exposure to security intrusions. While these measures are preferable to the damage that could occur from security violations, taken to their logical extreme they could eventually reduce the Internet to little more than a means of supporting certain pre-approved point-to-point data transfers.

To understand the evolution of this situation, we are carrying out a study to measure changes in Internet service-level reachability over a period of one year (to complete at the end of 1992). The study considers upper layer service reachability instead of basic network connectivity because the former indicates the willingness of organizations to participate in inter-organizational computing. The data we collect will allow us to compute relative network service growth among different countries, trends in the types of services to which sites limit access, types and geographical distribution of sites that limit access, and how sites limit service access to their network services.

The study consists of a set of runs of a program over the span of one to two days per measurement cycle, repeated every three months for a period of one year. Each program run attempts to connect to 13 different TCP ports at each of 12,865 Internet domains worldwide, recording the failure/success status of each attempt. The program attempts no data transfers in either direction. If a connection is successful, it is closed immediately. The machines on which connections are attempted are selected at random from a large list of machines in each domain, constrained such that at most 1 to 3 machines is contacted in any particular domain. The list of ports was chosen to span a representative set of services that can be expected to be found on any machine in a domain (so that probing random machines is meaningful). Only TCP ports are used, because they allow one to determine if a server is running in an application-independent fashion.

Clearly, a study of this nature raises a number of potential concerns regarding privacy, security, and network/remote site load. We have published a study plan that discusses these and other issues [17].

Conclusions

Resource discovery encompasses a range of problems that confront users of wide area networks in realizing the potential for remote collaboration and resource sharing. The Internet is a challenging environment in which to experiment with resource discovery, because of its scale, administrative decentralization, heterogeneity, and evolving commercialization.

In this paper we have discussed a number of aspects of the resource discovery problem, and efforts to address these problems in the context of the Networked Resource Discovery Project. In some cases, we indicated conceptual distinctions that are not always reflected in current system implementations. This is the case, for example, in the distinction between name-based and attribute-based systems. We also saw that ignoring conceptual distinctions in system implementations can sometimes burden users with extraneous information, and waste network bandwidth. This is case, for example, in the distinction between class and instance discovery.

Common to the Networked Resource Discovery Project efforts is the *Two Phase Discovery* paradigm, which treats resource discovery as a dynamic, exploratory process. In phase one, information that characterizes resources is extracted automatically and honed in quality through analysis of data semantics, context, and redundancy. Phase two distributes and searches this information in an efficient, flexible manner, based on where the information most naturally should reside. The dynamic nature of this paradigm is well suited to a large, decentralized, rapidly changing environment such as the Internet.

Resource discovery raises a number of privacy issues. We believe the tension between resource discovery and privacy is fundamental, and that a combination of policies and user awareness are needed to allow effective resource discovery without invading individual privacy.

Acknowledgements

This material is based upon work supported in part by the National Science Foundation under grants DCR-8420944, NCR-9105372, and NCR-9204853, a grant from Sun Microsystems' Collaborative Research Program, and a grant from AT&T Bell Laboratories.

Bibliography

Papers about the Networked Resource Discovery Project are available to Internet users by anonymous FTP from ftp.cs.colorado.edu, in the directory pub/cs/techreports/schwartz. The files in this directory are also available via an electronic mail interface. For more information, send a mail message to infosrv@ftp.cs.colorado.edu, containing the message body (not subject line) "send HELP" (without quotes). The author of this paper can be reached by electronic mail at schwartz@cs.colorado.edu.

1. V. Cerf. Networks. *Scientific American*, 265(3), pp. 72-81, September 1991. Special Issue on Communications, Computers, and Networks.
2. K. Harrenstien, M. Stahl and E. Feinler. NICName/Whois. Request For Comments 954, SRI International, October 1985.
3. CCITT/ISO. *The Directory, Part 1: Overview of Concepts, Models and Services.* CCITT/ISO, Gloucester, England, December 1988. CCITT Draft Recommendation X.500/ISO DIS 9594-1.
4. A. Emtage and P. Deutsch. Archie - An Electronic Directory Service for the Internet. *Proceedings of the USENIX Winter Conference*, pp. 93-110, San Francisco, California, January 1992.
5. B. C. Neuman. Prospero: A Tool for Organizing Internet Resources. *Electronic Networking: Research, Applications, and Policy*, 2(1), pp. 30-37, Meckler Publications, Westport, Connecticut, Spring 1992.
6. T. Berners-Lee, R. Cailliau, J. Groff and B. Pollermann. World-Wide Web: The Information Universe . *Electronic Networking: Research, Applications and Policy* , 2(1), pp. 52-58, Meckler Publications, Westport Connecticut, Spring 1992.
7. B. Kahle and A. Medlar. An Information System for Corporate Users: Wide Area Information Servers. *ConneXions - The Interoperability Report*, 5(11), pp. 2-9, Interop, Inc., November 1991.
8. M. McCahill. The Internet Gopher: A Distributed Server Information System. *ConneXions - The Interoperability Report*, 6(7), pp. 10-14, Interop, Inc, July 1992.
9. R. E. Droms. Access to Heterogeneous Directory Services. Proceedings of the Ninth Joint Conference of IEEE Computer and Communications Societies (InfoCom), June 1990.
10. M. F. Schwartz and P. G. Tsirigotis. Experience with a Semantically Cognizant Internet White Pages Directory Tool. *Journal of Internetworking: Research and Experience*, 2(1), pp. 23-50, March 1991.
11. M. F. Schwartz, D. R. Hardy, W. K. Heinzman and G. Hirschowitz. Supporting Resource Discovery Among Public Internet Archives Using a Spectrum of Information Quality. *Proceedings of the Eleventh IEEE International Conference on Distributed Computing Systems*, pp. 82-89, Arlington, Texas, May 1991.
12. D. J. Ewing, R. S. Hall and M. F. Schwartz. A Measurement Study of Internet File Transfer Traffic. Technical Report CU-CS-571-92, Department of Computer Science, University of Colorado, Boulder, Colorado, January 1992.
13. S. S. Coleman, D. C. M. Wood and M. F. Schwartz. Fremont: A System for Discovering Network Characteristics and Problems. To appear, Proceedings of the USENIX Winter Conference, San Diego, California, January 1993.
14. D. Hardy and M. F. Schwartz. Essence: A Resource Discovery System Based on Semantic File Indexing. To appear, Proceedings of the USENIX Winter Conference, San Diego, California, January 1993.
15. M. F. Schwartz and D. C. M. Wood. *Discovering Shared Interests Among People Using Graph Analysis of Global Electronic Mail Traffic.* Department of Computer Science, University of Colorado, Boulder, Colorado, February 1992. To appear, Communications of the ACM.
16. M. F. Schwartz. A Scalable, Non-Hierarchical Resource Discovery Mechanism Based on Probabilistic Protocols. Technical Report CU-CS-474-90, Department of Computer Science, University of Colorado, Boulder, Colorado, June 1990.
17. M. F. Schwartz. A Measurement Study of Changes in Service-Level Reachability in the Global TCP/IP Internet: Goals, Experimental Design, Implementation, and Policy Considerations. Request For Comments 1273, Department of Computer Science, University of Colorado, Boulder, Colorado,

November 1991.