Searching Internet Resources using IP Multicast

Hiroaki KASHIMA <kashima@csce.kyushu-u.ac.jp> *
Yoshiki ISHIDA <yoshiki@cc.kyushu-u.ac.jp> †
Zengo FURUKAWA <zengo@ec.kyushu-u.ac.jp> ‡
Kazuo USHIJIMA <ushijima@csce.kyushu-u.ac.jp>*

Abstract

The Internet is in a flood of information. Recently you can provide many kinds of information with several information providing systems. On the other hand, difficulty of finding out required information is increasing constantly, because of a huge set of data which are provided by many people on the Internet.

The following characteristics of the data cause the difficulty to find out the information. (1) The number of data is very large, (2) the data are widely dispersed on the Internet, (3) the mass of data is increasing rapidly, and (4) the data always change in structure and contents.

In this paper, a set of such data is represented as the word "super-dispersed information." It is difficult to find required information on super-dispersed information with existing information retrieval systems.

This paper introduces "March" which searches filenames on many anonymous FTPs using the IP Multicast. Multicast is a mechanism suitable to search super-dispersed information on the Internet. Filenames on anonymous FTPs are always changing and are dispersed widely. March consists of march-servers on anonymous FTPs and march-client on user machines. March is superior to archie systems in less traffic on the Internet and in shorter time for getting filenames. However, the current IP multicast restricts the March usability on user interfaces for specifying area of searching filenames.

1 Introduction

Finding objective information becomes difficult in the Internet. In the Internet, many kinds of information are widely spread and many systems are currently developed for providing information. These systems provide many data on computers and you can access the data through the Internet. However, finding objective data from many kinds of information provided by many kinds of information providing systems is difficult. In some cases you should search some kinds of information to retrieve objective data.

Existing information searching systems cannot search some kinds of information. They are designed for searching static data. The change of static data should be little or is managed completely. On the other hand, information providing systems, such as World-Wide Web[1], make information explosions on the Internet. These exploding data also change their structures and contents dynamically. The searching systems on static data cannot deal with volumes and transitions of exploding data.

We name such kinds of information "super-dispersed information," which is un-search-able or may become un-search-able because of the overflow of searching systems. Figure 1 shows the super-dispersed information. Super-dispersed information has some characteristics.

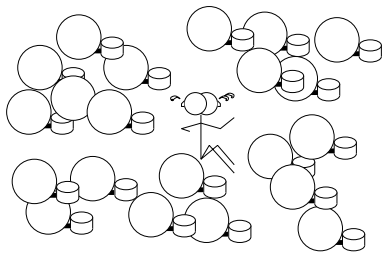

Figure 1: Super-dispersed information

- You cannot grasp all of existing information.
- There must be some leak of search result.
- Some information constructs a "weak" group. A weak group is to say the group

*Dept. of Computer Science and Communication Engineering, Kyushu University
†Computer Center, Kyushu University
‡Educational Center for Information Processing, Kyushu University

that people in it know such information in some ways.

- There are no precise boundaries among "weak" groups.

2 Existing Information Searching Systems

Existing information searching systems are not suitable for searching super-dispersed information. When you are going to search information from super-dispersed information, you should use an appropriate system to find information. Existing information searching systems have a common problem.

Systems that are usually called "information retrieval systems" are classified into three types: information providing systems, information searching systems and information searching-and-providing systems. Moreover, they are usually classified in another way into three types: centralized systems, tree structured distributed systems and net structured distributed systems. Table 1 shows the classification of information retrieval systems.

Table 1: Classification of Information Retrieval Systems

	centralized	distributed	
		tree	net
provide search	whois wais	DNS X.500	
provide	anon-FTP		WWW Gopher
search	archie Veronica		robot?

Centralized systems are not suitable to search super-dispersed information because the collection of super-dispersed information must overflow the server. Tree structured distributed systems are not also suitable to search super-dispersed information because these systems use the tree structured based on the relation of data but super-dispersed information has no precise relation. Net structured distributed systems are suitable in case the net can reflect the relations of super-dispersed information quickly enough to search.

Existing information searching-and-providing systems — such as whois, wais, X.500 and domain name system — are centralized or tree structured systems and they are not suitable to search super-dispersed information.

Existing information providing systems — such as World-Wide Web, Gopher and anonymous FTP — are centralized or net structured systems and data in these systems are weakly related or independent. Currently these systems provide super-dispersed information but these systems don't have the search method themselves.

Existing information searching systems — such as archie, Veronica and robot — are centralized or net structured system. Archie and Veronica are centralized systems and and they must overflow with data. Robot uses the net of the World-Wide Web but it transfers data at a stretch to the local host and searches on the local. Therefore the robot has heavy load on networks and World-Wide Web servers.

Super-dispersed information is provided by many information providing systems and existing information searching systems have several problems.

3 Searching Information with Multicast

Multicast[2] is a communication way suitable for searching super-dispersed information. Multicast is a communication protocol to send packets to members of a multicast group at the same time. You will send a query to the certain multicast group and you will receive search results from many searching servers which are members of the group.

3.1 IP Multicast

On the Internet virtual multicast backbone "MBone[3]" is running and many broadcasting experiments are done on MBone. MBone is constructed with LANs which have one or some multicast speaking hosts and tunnels which connect these LANs via non-multicast-speaking routers. This is based on Steve Deering's proposal in the latter of 1980's, which has following characteristics:

- Sender need not know all of the members belonging to the group.
- Multicast packets sent by a sender reach all of the members of the group.
- Sender need not join the group.

3.2 Use of Multicast for Searching Information

We propose a prototype here to search super-dispersed information efficiently. Figure 2 shows this prototype.

1. An information searching server stands with an information providing server and searches the data on the information providing server.
2. Information searching servers should join appropriate multicast groups based on the type of information provided there and on the place of the information providing server.
3. An information searching server need not exchange any data with any other information searching servers.
4. A user sends a query packet to appropriate multicast groups. The groups are determined based on the type of information and searching area.
5. An information searching server receives the query packet and searches data only which the information providing server have, and then sends back the search result.

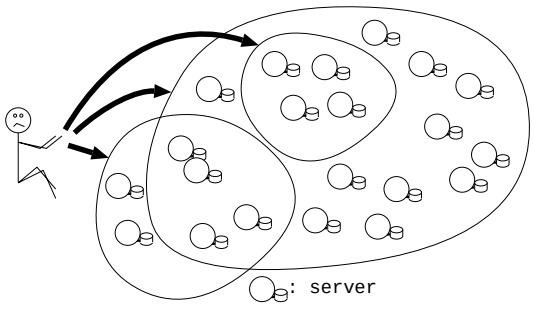

Figure 2: Searching super-dispersed information

With this method you will have following merits.

- You can know the closer place of information that you want in case you select appropriate groups.
- You can avoid to load many searching servers that do not have the types of information you want.
- Information searching servers should search data on only one providing server. Therefore the load of the searching server is light.
- You need not get all the data in the search area. Only a query and some search results are carried through the network.

3.3 To Use Current IP Multicast

The multicast group of current IP Multicast is not suitable for searching information. On the current IP Multicast you cannot use the multicast group to choose a specific area to which the packet reaches.

We have made a plan to select the multicast group with only the type of information and should select the search area with the TTL variable in the packet. The IP address of multicast group for searching information should be predefined, or you cannot select the group corresponding to the objective information. The TTL value should be decided when you begin searching. Each query packet will reach servers in the area which the TTL variable indicates.

4 File Searching System "MARCH"

There are too many files to search on many anonymous FTP servers on the Internet. They are a kind of super-dispersed information. You can use "Archie[4]" as a system to search files on many anonymous FTPs, but it has several problems. To solve these problems we have developed a new file searching system "March", which adopts the method we describe above.

4.1 Files on Anonymous FTPs

The anonymous FTP is an information providing system and the system itself has no method to search files on each server.

The number of files on anonymous FTPs is quite large and increasing rapidly. This is because an anonymous FTP server can provide many files with a large volume of disks. The cost of disks becomes cheaper and cheaper, then anonymous FTP server can provide more and more files.

Moreover, an administrator of an anonymous FTP server need not contact with administrators of other anonymous FTP servers. Therefore, anyone can start a new anonymous FTP server easily. For these reasons, files on anonymous FTPs are a kind of super-dispersed information.

4.2 Archie

"Archie" is a system for searching files on many anonymous FTPs. On the current Internet you can get the location of files that you want with this system.

The archie system is a centralized system. This causes a serious problem. The archie server must have filenames of many files on many anonymous FTPs, but the number of files on anonymous FTPs must become too many to manage and search on a single server because they are a kind of super-dispersed information. The archie system will overflow.

4.3 March

Searching files on anonymous FTPs with the above method is effective because files on anonymous FTPs are a kind of super-dispersed information. According to the above method, march servers have features as follows: each server corresponds to an anonymous FTP; each server searches filenames on the anonymous FTP; each server belongs to the march multicast group; servers exchange nothing each other; you can send a query packet to the march multicast group and receive search results directly from servers.

As described above, you select the march multicast group and you describe the searching area as the TTL of the query packet. It is useful to send query packets in order of the TTL from smaller to bigger because many of files are copies from the other anonymous FTPs. As there may exist many copies for the objective file, you can find it within a smaller TTL, and you can avoid making a load on servers over the specified TTL. At the same time, you can get the closest or mostly closest file to avoid making a load on networks. Figures 3 and 4 show difference between archie and march.

4.4 Implementation of March

The system is implemented in client-server model as shown in Figure 4. Each server joins the march multicast group and a client throws the query packet to the group. We use the IP address "224.224.2.2" for the march multicast group.

4.4.1 Server

The server is implemented on the SPARC SunOS4.1 with ipmulti-patch and is constructed with three part: the receiver, the searcher, and

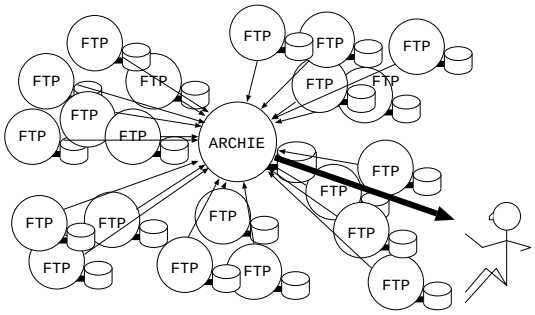

Figure 3: Search with Archie

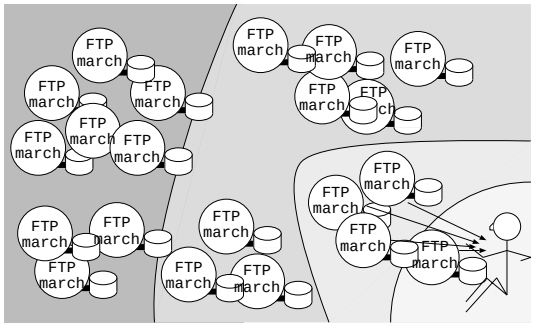

Figure 4: Search with March

the database builder. Figure 5 shows the function of each part.

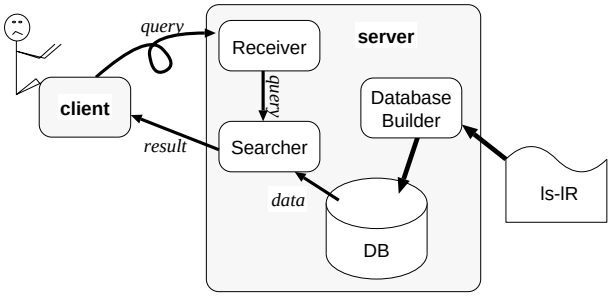

Figure 5: Construction of the March Server

The receiver runs as a daemon process and joins to the march multicast group to receive queries from clients. Each query is passed to the searcher. The receiver is written in C to use the functions of existing multicast applications.

The receiver makes the searcher start, which searches the filenames that match the query from the receiver. The filenames are compiled and stored in the local database built by the database builder. The searcher is written in Perl that is powerful at string processing.

The database builder makes periodically the local database from the `ls-lR` file on the anony-

mous FTP corresponding to the march server.

4.4.2 Client

The client is also implemented on the SPARC SunOS4.1 with ipmulti-patch and is written in C to use the functions of existing multicast applications. This sends a query to the multicast group with the appropriate TTL value. The query contains the query string, the search type and the display type. The client knows the searching servers dynamically with the ACK from servers. The client regards the receipt of search results from all senders of ACK as the end of searching.

When an objective file is not found, you can re-send the query with the larger TTL to search the larger area. Each server should discard the same query from the same client. Then the server should discard the same queries from the same host. We decide that the client must send the same queries from the same port and servers should discard queries from the same port of the same host.

4.4.3 The Search String

You can specify the search string at the beginning of the searching. If you don't, servers recognize the query as the request of ACK and you can get the server list.

4.4.4 The Search Type

You should also specify the search type, which indicates the type of the search string: "SUBSTR" as the substring, "REGEXP" as the regular expression and "SHELL" as the filename expansion method of the bourne shell.

4.4.5 The Display Type

You should also specify the display type, which is the request to servers for items and format to send. You should use "INFO" to request information of each file, or should use "URL" to request the URL of each file.

5 Evaluation of March

We experimented in MBone-JP in Feb., 1995. Seven organizations: Kyushu University, Kyushu Institute of Technology, Nagasaki University, Osaka University, The University of Electro Communications, Chiba University and Kyushu Area Regional Research Network started a march server of anonymous FTP on each organization.

5.1 Compatibility of March and MBone

The march system uses the virtual network MBone to transfer the query packet. However, MBone is mainly constructed for audio and visual conference. This causes a problem.

On the current MBone, you can find a file on an anonymous FTP connected with thick networks in preference to a file on an anonymous FTP connected with thin networks. The current MBone constructed to let packets flow in the thick networks in preference makes a query packet reach march servers connected with thick networks.

However, when you are on hosts connected with thin networks you must use large TTL to search out of the thin networks and then you cannot select proper searching areas except very large ones.

The result of our experiment shows this problem. We can search an objective file closer to us in Kyushu area. However, we can find the closest objective file out of Kyushu area with only searching almost all of servers in Japan.

This is caused by the current MBone. The current MBone is a virtual tunnel network and does not reflect the topology of the Internet. Figure 6 shows the current JP-MBone.

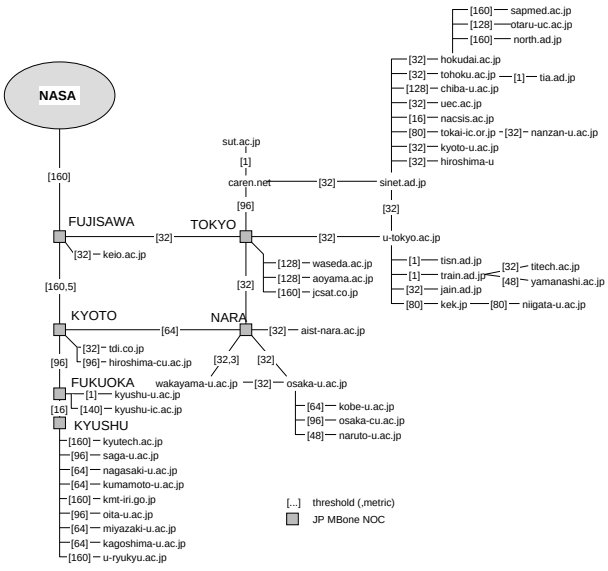

Figure 6: JP-MBone connection as of Feb. 2, 1995

5.2 Load of Network

With Archie, you will get the list of files that the server has and you cannot find the closest objective file. On the other hand, you can find the closest objective file with March. Table 2 shows the search result from Kyushu University with the search term "kterm" and the search type REGEXP.

Table 2: The number of found files with "kterm"

TTL	3	10	31	95	127	159	191
found file	4	2	0	3	22	0	20
newly found file	4	0	0	2	15	0	6
file total	4	4	4	6	21	21	27

We found total 51 files at the experiment, where 27 files were without overlaps. Many files were found when $TTL = 127$ and $TTL = 191$ but the number of newly found files was less on $TTL = 191$ than that on $TTL = 127$. 20 files were found on $TTL = 191$ but the number of the new files were only 6 and other 14 files were already found in the smaller searching area. We will be able to find the objective file with searching in the grade of organization or region if almost all of anonymous FTPs are search-able with March.

5.3 Searching Time on a Server

The searching time on a server depends on the size of the local database which depends on the size of the file `ls-lR` on the anonymous FTP. Figure 7 shows the relation between the size of `ls-lR` and the searching time. We measured the searching time with the search string "term" and the search type "SUBSTR", and timed on the HelioStation1030 (SPARC 2 clone) with the 80MB memory.

The searching time is in proportion to the size of `ls-lR` but there are two lines that have different slants. The upper line is expressed by $t = 1.98 \times 10^{-3} \times x + 1.52 \times 10^{-3}$ and the lower one is expressed by $t = 7.35 \times 10^{-5} \times x + 6.15 \times 10^{-2}$. We are investigating the reason why the relation between the searching time and the size of `ls-lR` is approximated with two lines.

5.4 Influence of Increase of the Number of Files

The increase of the number of files on anonymous FTPs is due to both the increase of the number of files on an anonymous FTP and the increase of the number of anonymous FTPs.

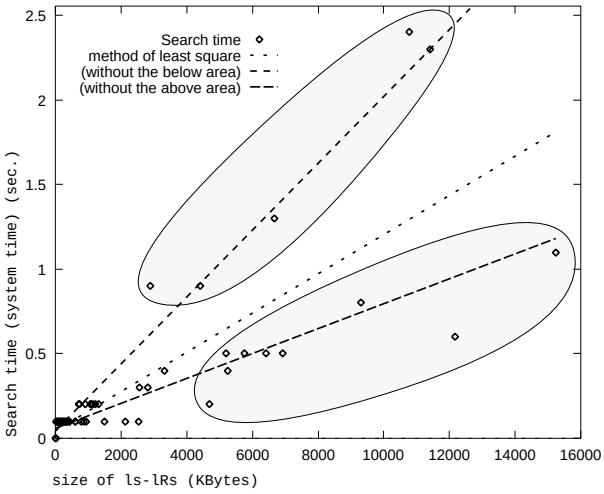

Figure 7: ls-lR file size and searching time

The increase of the number of files on an anonymous FTP should not make longer the search time of the local database. The increase of the number of files on an anonymous FTP will enlarge the local database. However, the search time of the local database will not be longer since computers will become more powerful.

The increase of the number of anonymous FTPs should not make the search time longer, too. The number of march servers will increase as the number of anonymous FTPs because each march server stands with the corresponding anonymous FTP. The increase of the number of march servers will bring following results.

- The file which you want will be found with a query to the narrower area.
- The time to get the search result will not be longer since it depends on each server's searching time.
- Each server's load can become heavier but this matter is avoidable. When the number of servers in a certain area will increase, a query to the area will make a load on more servers. However, at the same time, you can avoid unnecessary loads with narrowing the search area because you will find the objective file with searching the narrowed area.

In comparison, the database will be too large to have on the archie server because the database becomes larger in proportion to the increase of total number of files on anonymous FTPs.

The march system can correspond to the increase of the number of files with which the archie system cannot deal.

5.5 Application to Other Systems

The World-Wide Web and the Gopher are the other typical super-dispersed information providing systems. We consider the searching of these kinds of information with the method.

To search information on the World-Wide Web, the robot system reflects the relation of data on the World-Wide Web better. However, the robot system makes a load on many servers by getting many data one after another. In addition, this system makes a load on the network by transferring whole data to the machine that the system works for searching. Super-dispersed information should be searched with a dispersed system of searching information.

To search information on the Gopher, the Gopher net is useful for searching like the robot, but a system to search on each Gopher server is required. Veronica system is to search Gopher space, but this system will overflow like archie because this system is centralized, too.

There is another problem. There are no copies on both systems but IP Multicast can lose packets. Therefore you may not find what you want.

In addition, you should be careful not to insert security holes when you implement a searching system with the method. You should use the method only for searching information.

6 Summary

We have proposed a searching method of super-dispersed information using the IP Multicast. Super-dispersed information is some kinds of information that have difficulty to search and that may overflow information searching servers. Existing information searching systems are not suitable to search super-dispersed information.

This method has the following merits: you can know the location of information closest to you; you can avoid to make a load on many searching servers that do not have the types of information you want; each server's database is small and the server's load is light.

We have developed the file searching system "march." This system uses the IP Multicast to send a query to many servers at once. Servers' load is light because each server has the filenames on only an anonymous FTP. A user can search files in several areas using the mechanism of the IP Multicast.

Our future work is as follows:

1. to propose and implement the flexible multicast group that can be useful for searching information,
2. to propose and implement the reliable multicast transfer method to avoid the failure of searching due to the loss of query packet, and
3. to apply the method to the several types of super-dispersed information.

Acknowledgment

The authors would like to thank researchers of WIDE Project and JAIN Consortium for discussion, and staffs of Nagasaki University, Kyushu Institute of Technology, Osaka University, Nara Institute of Science and Technology, The University of Electro-Communications, Chiba University, and KARRN Associations for experiments and discussions.

References

[1] T. Berners-Lee, R. Cailliau, J-F. Groff and B. Pollermann. "World-Wide Web: The Information Universe." *Electronic Networking: Research, Applications and Policy*, 2(1):52–58, Spring 1992.

[2] S. Deering. "Host Extensions for IP Multicasting." Request for Comments 1112, August 1989.

[3] H. Eriksson. "MBone – the Multicast Backbone." in *INET'93*, August 1993.

[4] A. Emtage and P. Deutsch. "archie - An Electronic Directory Service for Internet." in *Proc. of the Winter 1992 USENIX conf.*, 1991.

Author Information

Hiroaki Kashima received M.Eng degree in computer science and communication engineering from Kyushu University, Fukuoka, Japan. and now works for Fujitsu Co. Ltd.

Yoshiki Ishida is a lecturer of computer center at Kyushu University. His research interests include mobile communications and collaboration

over the Internet. He is a member of Internet Society, IEEE.

Zengo Furukawa is an associate professor of educational center for information processing at Kyushu University. He received Dr. of Eng. in computer science and communication engineering from Kyushu University. He is a member of the IEEE Computer Society, ACM, IPS Japan, IEICE Japan, and JSSST.

Kazuo Ushijima is a professor of computer science and communication engineering at Kyushu University. He received B.Eng. and M.Eng. from University of Tokyo, and received Dr. of Eng from Kyushu University. He is a member of the IEEE Computer Society, ACM, IPS Japan, IEICE Japan and JSSST.