Advertising Peer-to-Peer Networks over the Internet

Matthieu Weber[1], Jarkko Vuori[2], and Mikko Vapa[3]

[1] University of Jyväskylä (mweber@mit.jyu.fi)
[2] University of Jyväskylä (Jarkko.Vuori@jyu.fi)
[3] University of Jyväskylä (mikvapa@it.jyu.fi)

Abstract. Most of the peer-to-peer overlay networks either provide a centralized way to join the network or rely on out-of-band methods for that purpose. None of them is satisfactory, since the first one relies on a single point which can fail, thus making the network vulnerable to failure and the second one gives no solution to the problem. This document proposes a new approach to the problem, using already existing systems such as Usenet, IRC and Web search engines to advertise the presence of an overlay network on the Internet and thus facilitate the binding of new nodes into such a network, in a fault tolerant way.

1 Introduction

Peer-to-peer (P2P) computing [1] has become in a couple of years a hot topic. Since the success of Napster [2], many systems based on P2P overlay networks have been developed, mainly in the field of file sharing. Although actually working systems provide a mechanism allowing new nodes to join the network, most of the efforts in the Research World are concentrated on locating and retrieving resources from the network and most of these systems consider that the acting nodes are already part of the network.

Some systems, like Napster or ICQ [3] are using central indexes (sometimes called "databases" or "directories") referencing all current users of the system in order to allow new users to locate other peers and thus communicate with them. The mean of accessing the index (usually an IP address and a TCP or UDP port number) is known before using the system (e.g. from a configuration file given with the software), and thus represents an easy way to join the network. Recent history has proven that this central index is a potential point of failure and performance bottleneck, and that bringing down the index will in time bring down the entire system: current users are still members of the network, but they can't rejoin it after leaving, and no new user can join the network. In time, the network will disappear. Backup servers are not always a solution in this case: if a company is running the servers (like e.g. Napster) is stopped by a decision of justice, the backups are usually stopped at the same time. Similarly, if only a limited set of backups exist, it is possible to stop the Internet traffic to and from those servers to censor the whole system.

In order to avoid this single point of failure, fully distributed P2P systems like Gnutella [2] have been developed. However, when a P2P system is designed in such a way that there is no central control over the network, users must find a way to join that network prior to being able to use the system. But finding an entry point into the network becomes a problem when one doesn't know where to start looking for it.

In this paper, we will discuss the existing systems that try to address that problem, and sketch the design of a system that allows finding peers of any network using the already existing infrastructure of the Internet.

2 The Real-World Example

A hint of a solution to that problem can be found in the real world. When one person wants to know about a given service, there are several ways to find information about that service, using well-known sources of informations, i.e. sources that are general knowledge. Each of those ways can be transposed in the world of P2P systems:

- Ask friends if they know something about the service: this is equivalent to already being member of a P2P network.
- Look into the phone book, and especially in the yellow pages, into the local newspaper or into specialized magazines for advertisements: this can be compared to using an already existing, well-known system, like a search engine (e.g. Google [4]), Usenet News [5] or IRC [6].
- Knock randomly on people's doors, until one find the required information: this can be assimilated to random network scan.

The first example is not possible, since the user is not yet a member of any P2P network, but the two other examples are worth being investigated.

Finding a source of information is one more step that must be taken in order to find the desired service, and which only displaces the problem. However, the fact that the source of information is *well-known*, i.e. part of the *general knowledge*, makes this step possible for anybody.

3 Existing Systems

Existing systems already try to address the problem of entering the network. Several solutions have been found, that can be divided into three categories, depending on the way their resource index is organized [7]:

- Systems with centralized indexes, where a list of potential entry points are available from one source only.
- Systems with decentralized indexes, where the lists of entry points are scattered over several sources but where the location of those sources is to be found from well-known places.
- Systems with no index, where the lists of entry points are directly available from well-known places.

3.1 Centralized Indexes

Napster[2], Gnutella [2] at some point of its existence, Kazaa [8] and eDonkey [9] fall in the first category: the central index of Napster and Kazaa, maintained by the companies developing the softwares, and the early Gnutella cache [2, p. 113–115] are the well-known sources of information, and eDonkey distributes a list of potential entry points along with their software distribution. Those central points are the well-known sources of information, but also potential sources of major system failure.

3.2 Decentralized Indexes

More recent configurations of Gnutella [10] belong to the second category: there are several independent lists of peers available from the World Wide Web (so if one list disappears, there are still several others), but finding those lists depends on a Web search engine. There are however several major and independent search engines available (so this is not either a potential point of failure) representing the well-known sources of information. The risk of global failure is smaller than in the previous case thanks to this redundancy. However, the use of those lists still requires human intervention, because of the need for manipulating a Web search engine (section 4.3 discusses the automatization of that task), and when one list disappears from the Web or moves to another location, it can take a couple of months until the index of the search engine is updated.

3.3 Without Index

JXTA [11], the Universal Ring [12], GIrcCache [13] are in the third category: the needed information is available from numerous well-known sources (RendezVous servers [11] of JXTA, each node of the Universal Ring, the local IRC server in GIrcCache). Once one source is known, the other ones can easily be found through the first one. The risk of failure is here low, thanks to the high redundancy of the well-known sources of information.

JXTA is however only dodging the issue: all JXTA users are members of one peer group called the World Group, and joining this group in order to know other RendezVous servers requires to know a first RendezVous server, hosted by Sun Microsystems, which is hard-coded in the software distribution of JXTA. Because of that, as long as JXTA is not widely used (i.e. the address of the closest RendezVous server is a well-known information for everybody), it will actually fall into the category of centralized systems. In the same way, the Universal Ring expects the system to be widely distributed, and that each user uses a well known node as entry point.

GIrcCache, on the contrary, is really fully distributed: it relies on IRC, which is a 10 years old, well established and fully distributed chat system. Accessing GIrcCache and finding Gnutella nodes only requires to know the local IRC server, which is assumed to be a well-known information (as are the Internet address of the local Mail server and Domain Name server). GIrcCache however allows to find only Gnutella nodes.

4 Towards a Universal Solution

The existing solutions presented above are designed for being used within a specific system (except the Universal Ring), and often assume that the system is already sufficiently widely used in order for the source of information to be well-known. The latter is however not true, and, although the Universal Ring is one possible solution for the future, none of those systems is nowadays widely spread.

A transitional solution needs to be set up so that one (or maybe several) system can in future become as usual as the e-mail, the Web, Usenet or IRC are nowadays. This transitional system should be widely available as soon as possible, without the need of heavy infrastructure deployment, in order to make it attractive to the peer-to-peer software developers. One easy way to achieve this is to rely on already existing infrastructure, which are already part of the general knowledge of the Internet. Usenet, IRC, and, to some extent, the Web can be used in that way.

4.1 Usenet

Usenet [5], or sometimes called Internet News, is a distributed, completely decentralized system that broadcasts messages across a large set of servers. As Usenet is as old as the Internet itself, it is considered as one of the basic services of the Internet, and should be available from any Internet Service Provider. However, given the absence of central administration in Usenet, getting an accurate picture of how many Usenet servers are available in the world is difficult. Some Web sites propose lists of public servers [14, 15], but there is no way to infer from those data the availability of Usenet servers for everybody. It is however assumed that most of the people having an access to the Internet are also given an access to a Usenet server by their Internet Service Provider.

Messages in Usenet are sorted by topic into newsgroups, basically one newsgroup for one topic. Some groups, however, have special roles, like informing Usenet server administrators about newly created groups and groups that must be destroyed, discussing about whether a group should be created or destroyed, and test groups, where users can verify that their news reading softwares work properly (especially for posting messages to a server). The content of the test groups is also propagated among servers, in order to check if propagation works properly, but no user actually reads it.

It would thus be possible to use those test groups to post adequately formatted messages describing a peer-to-peer network and a node which is a member of that network, in order to allow other users to find an entry point into that network (the format of that message will be discussed later).

As an alternative to using test groups for that purpose would be to create a dedicated group in the alt.* hierarchy. This solution might be cleaner, but if alt.* groups can be easily created, they can also easily be destroyed (actually ignored by administrators), thus making the whole system vulnerable to attacks.

4.2 IRC

IRC (Internet Relayed Chat [6]) is a real-time chat system based on a network of servers which are exchanging messages in a peer-to-peer way, without any central control. Users connect to their favorite server to join the system. They are then uniquely identified by a nickname, and can send messages either to another user or to a channel. Channels are virtual rooms, which the users can enter or leave. Every message sent to a channel is forwarded to all members of that channel; it can thus be assimilated to a multicast group. Any user can create a new channel, simply by "joining" a non-existing one.

It is thus possible for some nodes of a peer-to-peer network to connect to IRC in order to find other nodes to whom they could connect. This is already implemented in GIrcCache [13] for the Gnutella network, but we propose here to extend this to any network and to design a lightweight protocol for that purpose.

4.3 WWW Search Engines

GWebCache [10] relies partly on the WWW search engines, which represent the well-known sources of information for finding GWebCache servers. A search engine is by essence a centralized system, which is subject to failure (in this case, a failure can be an attack, but also the disappearance of the company running the search engine or censorship). As there are several search engines available on the Web, this problem can be somewhat minimized, but not completely forgotten.

However, a search engine can be relied on in case none of the above-mentioned systems is available or known by the user.

4.4 Random Network Scanning

If none of the above solutions is available for a given user, the last resort possibility would be to try to connect to random IP addresses, until a peer is found. This solution should however be avoided, since this behavior can be assimilated to network scanning, which is considered as an attack by some network administrators.

5 Architecture Design

If we consider that Usenet, IRC, a search engine and random connections are several data communication media, we can design a two-layer system (Figure 1) composed of an Information Layer and a Transport Layer, running on the top of the various media which would allow a peer-to-peer application to use all of them in a transparent way.

We also describe basic considerations on the behaviors that must be observed by the peers when using those media in order not to overload them: IRC and Usenet were not designed for the purposes described in this paper, and a massive use of these systems by peer-to-peer networks may rapidly overload them. The

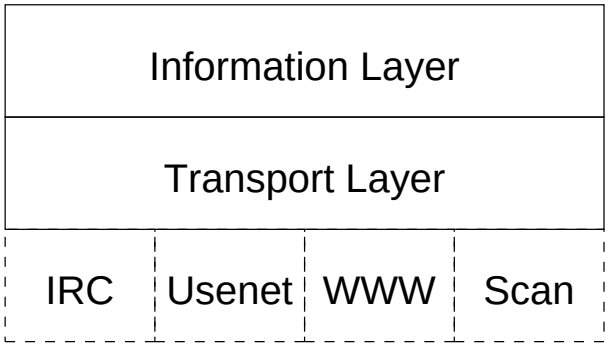

Fig. 1. Protocol Stack

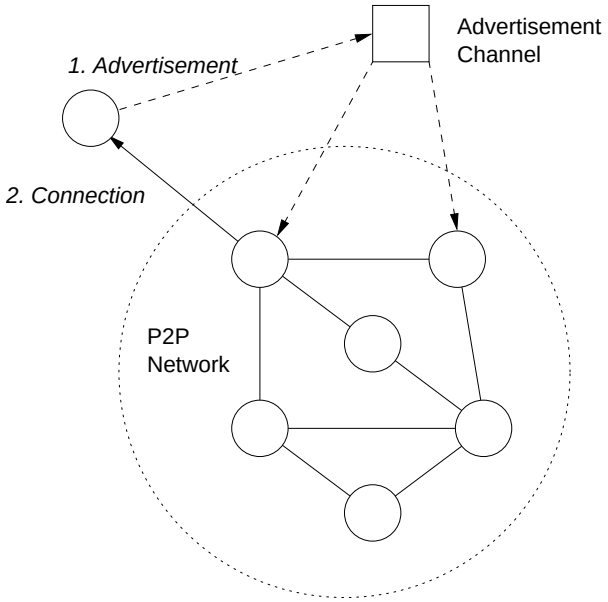

Fig. 2. Joining Mechanism

greatest care must thus be taken when implementing this system to minimize the risk of provoking a denial of service in these systems.

The system is designed for advertising a node's presence and its will to enter a network. It thus does not allow a node to directly discover other nodes of that network, but on the contrary to be found by the other ones (see figure 2). This expected behavior relies on the assumption that nodes which are already members of the network will be listening to those advertisements and initiate connections to the new nodes. This paper also assumes that some nodes which are already members of the network but willing to get new neighbors will listen to the various media.

5.1 Information Layer

When the node wants to join a network, its own binding information is sent over the Internet through the Information Layer.

The Information Protocol is XML based and has only one primitive: the *ad* primitive. Here is the Document Type Definition of the XML messages of the Information Protocol:

```
<!ELEMENT ad (network,bind-data)>
<!ELEMENT network ANY>
<!ATTLIST network
          name #CDATA REQUIRED >
<!ELEMENT bind-data ANY>
<!ATTLIST bind-data
          type #CDATA REQUIRED >
```

The messages are designed so that almost any kind of information can be sent. The only mandatory information is the name of the network which identifies it among all peer-to-peer overlay networks (and which must be decided by the designers of the network; no name-conflict resolution mechanism will be provided) and the type of binding information contained in the message (in the case of the Internet, the type will be *TCP/IP* and the *bind-data* element could contain an *ip* element and a *port* element. In the case of Bluetooth or IrDA protocols, these information may be completely different.)

ad messages are then given to the Transport Layer, which will multicast them to any peer which is willing to receive them, according to the media which are available to the Transport Layer.

ad messages can also be received from the Transport Layer. The Information Layer must forward to the upper layer only information relevant to the desired network. All other information is dropped.

5.2 Transport Layer

The Transport Layer will send the messages it receives from the Information Layer over all the media it can access: Usenet, IRC, Web search engine, random

network scanning. The messages will be fit into a suitable form, depending on the media.

Media are assigned priorities by the user who administrates the node running the software: the medium having the highest priority is used first. If it doesn't yield any result after a given timeout, the next medium in the priority list is tried (this does not necessarily mean that one gives up the previous one). If no result has been reached at this point, joining the network doesn't fail, but is considered as being delayed. Once all media have been given up, the network connection fails. It is however possible to remain connected to a given media for days (e.g. IRC) or to poll information at regular intervals over a long period of time (Usenet, Web search engine). Joining a network might thus take a lot of time, but failure will occur only if none of the media is available or if the user interrupts the procedure.

IRC Medium Before sending a message using the IRC medium, a TCP/IP connection must be made to an IRC server. The address and port number for establishing that connection is considered as common knowledge and is given by the user who administrates the node running this software. Once the connection has been established, the *#p2padvertisement* channel will be joined and one only message will be sent to the channel, providing the users already connected to the channel with the binding information of the current node.

All messages sent by the other nodes over that channel will be forwarded to the Information Layer.

The use of IRC medium gives a naturally limited lifetime to the advertised information: messages will only be visible to the nodes already connected to the channel at the time of the sending of the message. There is no need to re-send the message: when a new node arrives, it sends its own information, and one can react on it (i.e. try to establish a connection with it) if it is found suitable.

Usenet Medium Before sending a message using the Usenet medium, a TCP connection must be made to an Usenet server. As in the IRC medium, the address and port number for establishing that connection is considered as common knowledge. Once the connection has been established, the *alt.test* newsgroup is to be requested and some message headers are to be read (this group usually has a lot of traffic, so reading the messages are limited to the last N ones, N being given as a configuration parameter by the administrator of the node). Only the messages whose *Subject* field is set to "p2padvertisement" will be read (i.e. their bodies will be requested from the server). The bodies of those messages are then passed up to the Information Layer. If no suitable information is found, the message coming from the upper layer is posted to the newsgroup.

Usenet servers associate each message with an index number which is growing monotonously. It is advisable to cache the last index number which has been requested, in order not to read the same messages again during the next poll.

The use of the Usenet medium gives a naturally limited lifetime to the messages: in order to save disk space, Usenet servers delete older messages. The

usual lifetime of a message is between a day and a month, depending on the newsgroup and on the server.

Web Search Engine If IRC and Usenet are not available at a given location (because the local network administrator is blocking the necessary TCP ports with a firewall), the WWW might still be reachable, even if its use as a broadcast medium is more complex.

Compared to the two previous media, the use of a Web search engine requires an additional Web server for publishing the binding information. Moreover, different web search engines have different user interfaces, which require specific programming interfaces.

However, the use of a Web search engine as a diffusion medium differs little in principle from the Usenet medium. Before sending a message using this medium, a TCP/IP connection must be made to the Web server hosting the search engine. The address and port number for establishing that connection is considered as common knowledge and is given by the developer who made the programming interface. Once the connection has been established, a request can be sent using HTTP protocol and the resulting HTML page can be parsed in order to find URLs of Web pages containing binding information. Each of those pages must then be parsed to get the necessary information, which can finally been passed to the upper layer.

Advertising the binding information is however less straightforward than in the previous cases: it must be published on the Web, in a form that is easily exploitable by search engines, and the Web document must be registered into the search engine. Simply publishing an Information Protocol message might not work with all search engines, since the message is in XML, not HTML. However, the registration needs to be done only once for a node.

Data Format Here is the way to publish the binding information on the Web that is proposed:

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD HTML 4.01//EN"
  "http://www.w3.org/TR/html4/strict.dtd">
<html>
 <head>
  <title>p2padvertisement</title>
 </head>
 <body>
  <p><a href="binding-data.xml">
    binding-data.xml</a></p>
 </body>
</html>
```

The actual binding data is located in the `binding-data.xml` file, to which the HTML document links. The *title* element of the HTML document is mandatory

and used as a filter criterion for selecting the results given by the search engine. Maintaining the information up-to-date consists in maintaining the content of the `binding-data.xml` file up-to-date.

One can notice that the use of Web search engine medium does not give a natural lifetime to the binding information. This information must be kept up-to-date, by means which are out of the scope of this document.

Drawbacks Compared to the previous media, the use of a Web search engine as a diffusion medium for binding information is not thought to be a good choice for automated network binding, for the following reasons:

- No standard programming interface between search engines (the layout of the result page of search engine A is different from the one of search engine B).
- No standard programming interface within one search engine (the layout of the result page can change at any time without warning).
- The request might yield unwanted results, which make the parsing of the result page less efficient.

However, some search engines might provide an automaton-friendly search interfaces (like the SOAP [16] interface of Google [17]). Some others might be developed in the future.

Random Network Scan If all previous media fail to return the desired information, or if the user who installs the software is not able to give any of the needed information (IRC/Usenet server IP address, Web search engine interface), the last resort solution is to connect to some random or pseudo-random IP addresses in order to find a peer to which to connect.

This medium is not a diffusion medium in the sense that it advertises the node's information to other nodes, but it is nonetheless potentially able to return binding information to the upper layer.

Heuristic methods that can be applied are, among others:

- Trying IP addresses of nodes to which the current node was connected earlier (this requires for the node to keep a history of its neighbors).
- Trying IP addresses of the same subnetwork than the current node.
- Trying random IP addresses inside the network of great domestic or foreign institutions (e.g. universities. A list of these institutions is asked from the user).
- Trying random A class IP addresses from big Internet Service Providers (these addresses or networks is asked from the user).

None of those methods is might yield satisfactory results, and they can even be considered as illegal in some countries. The use of this medium should be disabled by default, and enabled only on the user's request.

6 Conclusion

This paper describes a way to advertise peer-to-peer overlay networks over the Internet in a fault tolerant way, using the already existing basic services of the Internet (Usenet, IRC, Web search engines) and one backup solution in case the previous are not available (random IP address scan).

The goal of this system is to facilitate joining a peer-to-peer overlay network when one does not yet know any member of that network, using already available media such as the ones above-mentioned in order to advertise the information required for joining the network, and avoiding the need to develop specific infrastructures. This solution could be used as a temporary solution, until a more specialized and more efficient infrastructure is developed.

IRC and Usenet media are considered as more efficient and easier to use than Web search and random IP scan, and are thus preferred.

Further work will focus on developing a prototype application implementing the principles described here and testing it with Chedar, a peer-to-peer experimentation platform developed at the University of Jyväskylä.

References

1. Schollmeier, R.: A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In: Peer-to-Peer Computing, München, Germany (2001)
2. Oram, A., ed.: Peer-to-Peer: Harnessing the Power of Disruptive Technologies. 1st edn. O'Reilly & Associates, Inc. (2001)
3. Rein, L.: O'reilly network: Icq. `http://www.oreillynet.com/pub/d/572` (2001)
4. Google Inc.: Google. `http://www.google.com/` (2003)
5. Salzenberg, C.: What is usenet. `http://www.faqs.org/faqs/usenet/what-is/part1/` (1999)
6. Oikarinen, J., Reed, D.: Internet relay chat protocol. RFC 1459, IETF (1993)
7. Arturo Ribeiro, O., Weber, M.: P2p application's architectures. Technical report, University of Jyväskylä, Agora-Center (2003)
8. Sharman Networks `http://www.kazaa.com/us/help/glossary.htm`: KaZaA. (2002-2003) Version 2.1.
9. MetaMachine `http://www.edonkey2000.com/documentation/index.html`: eDonkey2000. (2002)
10. Dämpfling, H.: Gnutella Web Caching System, `http://www.gnucleus.net/gwebcache/specs.html`. (2002)
11. Gong, L.: Project jxta: A technology overview. http://www.jxta.org/ (2001)
12. Castro, M., Druschel, P., Kermarrec, A.M., Rowstron, A.: One ring to rule them all: Service discovery and binding in structured peer-to-peer overlay networks. In: SIGOPS European Workshop. (2002)
13. Anonymous: Gnucleus. `http://www.gnucleus.com/` (2002)
14. Anonymous: Newzbot! public usenet resources for the masses. `http://www.newzbot.com/` (2003)
15. Anonymous: Usenet top1000 servers. `http://www.top1000.org/` (2003)
16. Box, D., al.: Simple object access protocol (soap) 1.1. Note, WWW Consortium (2000)
17. Google Inc.: Google web apis. `http://www.google.com/apis/` (2003)