Source Discovery versus Sink Discovery in P2P: A Manifesto

Jason Rohrer
University of California, Santa Cruz
Department of Computer Science
Santa Cruz, CA 95064
+1-831-429-4294
rohrer@cse.ucsc.edu
http://jasonrohrer.n3.net

December 9, 2002

1 Source discovery: The dominant paradigm

We can view a content exchange system as a graph with a node for each user and an edge for each connection. Some nodes in the graph are content sources—these nodes have content that other people want. Other nodes are content sinks, since they want content that other nodes have. Of course, when we consider all of the content exchanged in the network, many nodes are both content sources *and* content sinks. However, for a given piece of content, we can assert in general that each node is a source, a sink, or neither, but never both a source and a sink—people rarely seek a piece of content that they already have.

Almost every content exchange system—from the web, to email, to television—can be modeled in this way. Nearly all internet-based content exchange system can be pigeonholed even further: they are *source-discovery* systems. By source discovery, we mean that content sinks are the active parties. Sinks seek out sources with the content that they want, and sources passively wait for sinks to contact them. The web, augmented with Google [1], is a source-discovery system. Gnutella [2] is a source-discovery system. Freenet [3], FastTrack [4], Chord [5], edonkey [6]—all are source-discovery systems in one way or another.

Are there any exceptions to the sovereignty of source discovery on the internet? One exception comes to mind, but mainly in terms of its underlying implementation: Usenet newsgroups [7]. When a Usenet server has a newly posted article to share with the world, it contacts its neighboring servers to discover which ones want the article (in other words, which ones are subscribed to the article's newsgroup). The server with the new article is a content source, and other news servers around the world that are subscribed to that newsgroup are the corresponding content sinks. From the end-user's point of view, however, Usenet might

seem like a source-discovery system: users contact their news server to download the latest news articles.

1.1 Implementations percolate into user-space

Underlying implementations are interesting in their own right, but do the end-users really care? How can an end-user tell the difference between a source-discovery system and a sink-discovery system anyway? The hallmark of source-discovery systems is *search functionality*. Regardless of underlying implementations, when a user performs a search, he or she is actively looking for content, and the content is sitting there, waiting to be found. Notice that Usenet has no built-in search functionality. Despite the fact that you actively contact your Usenet server when you want your news, the news essentially finds you. Email is similar in this regard: your messages find you, and the source (the sender) is the active party.

With the examples given so far, it might seem like the internet is populated with both source- and sink-discovery systems. Perhaps each model is a good fit for certain types of applications. However, we point out that source discovery is the dominant internet paradigm. When someone sets out to design a content exchange system for the internet, unless something specific about their application forces them toward another model, they build a source-discovery system.

So what? What's wrong with the source-discovery model anyway?

1.2 Source discovery in the real world

When we look at the non-internet world, we see very few source-discovery systems of any kind. In particular, we can see that content exchange in the real world is not based on source discovery.

Consider the publication model. An author writes a book, and that book is distributed by the publisher throughout the market. The publisher uses some kind of heuristic to determine an appropriate density for this distribution (for example, more copies of the book are shipped to regions that have bought many copies of the author's previous books). The author is the source, the readers are the sinks, and the books are the units of content. Imagine source discovery in this example: each reader must travel to the author's house in order to obtain a copy of the book. Sounds absurd, right?

Source-discovery-based content exchange systems do not occur in the real world because they are a pain in the neck. People probably figured this one out shortly after they invented content and tried to exchange it. However, using all of the modern wizardry of computers and the internet, we can throw centuries of real-world wisdom out the door.

You want content? You find it.

1.3 The abominable burden of searching

Searching is a painful process.

You pick words. You type them in. You click "search." You get results. The results are not what you are looking for. You pick new words. You type them in again. You click

"search" again. The new results are better. You fetch the top-ranked result. Still not quite what you are looking for. You pick new words. You type them in...

Before Google (yes, kids, when we were growing up, there was no Google), searching was atrocious. Now that we have Google, searching is merely abominable. But Google only works for one system, the web. Scores of other source-discovery systems rely on search as their central mechanism, and these other systems do not have Google. In a system as dynamic as Gnutella, it is debatable whether a Google-like service would even be possible.

So, users of other systems are faced with the atrocious burden of searching. We have all wasted precious hours of our lives searching and re-searching. We have all abandoned particular searches with groans of frustration after finding nothing that we wanted, even though *everything* is surely out there.

Rest assured: you can put the traumatic memories of failed searches behind you. There is an alternative.

2 Sink discovery: Reversing the dominant paradigm

When we take the dominant source-discovery model and flip everything upside down, we come up with what we call the *sink-discovery* model. Sinks, those that want a piece of content, are passive. Sources, those with the content, actively seek out sinks for their content. Jumping back to our real-world example, an author writes a book and uses a publisher to distribute it to those that want it. Readers do very little work other than traveling to their local store to pick up their copy.

Sink discovery in the real world is well and good, but how can we translate this into an internet-based content exchange system? How can sinks define what they want? How can sources figure out who wants what? And how in the world will the distribution work?

Many models are possible for enabling sinks to define what they want and for enabling sources to fulfill those wants. A channel-based model is perhaps the simplest, and we will focus on it here. Sinks can subscribe to various channels, and sources distribute appropriate content on each channel. A sink will automatically receive everything sent out on the channels to which it is subscribed.

What about distribution?

Returning to our author/book/publisher example, we can take the publisher completely out of the picture—on the internet, *everyone has a printing press.* Once the author distributes copies of the book to a few readers, those readers can then become sources for the book's content. In other words, the author contacts a handful of readers directly and gives each one a copy of the book. Each reader then makes copies for a handful of other readers, each of which makes copies for a handful more, and so on. Throw in a dash of six-degrees-of-separation socio-mythology, and before too long, everyone who wants a copy of the book has a copy. This example can be translated directly into our channel-based model: all of the readers are subscribers to the author's channel.

In terms of discovery (in other words, how people that have the book find people that still want the book), the mechanisms used for source discovery in existing systems can be inverted and used directly for sink discovery. For example, a Gnutella-like search query

("who has this?") can be transformed into a sink-discovery query ("who wants this?") and routed in the opposite direction.

2.1 Spam

Did we really write "channel-based"? The only channel-based system that made it is Usenet, and it is now on its last legs because of spam. Usenet's fatal flaw is that essentially *anyone* can publish articles on *any* channel *without accountability*. Also, email can be thought of as a channel-based system in which we each subscribe to a unique channel. Again, the fatal flaw is present: essentially anyone can email anyone else without accountability.

We are not going to advocate a channel-based system without dealing with accountability head-on. When we wrote "the author's channel" above, we meant it: only the author can send out books on the author's channel. When readers get books on that channel, they can be sure that the author sent them out. This kind of accountability can be guaranteed easily using modern encryption techniques: the author can digitally sign everything that goes out on the author's channel.

2.2 Extending the scope

So far, sink discovery sounds great for authors that want to send out their latest books, but not useful for much else. We can switch gears now and leave our real-world example for the moment. Instead of authors and books, we can talk about channel owners and generic content. Nothing forces channel owners to distribute content that they themselves authored, and nothing forces them even to distribute content that their subscribers actually want. Of course, since all content is signed, channel owners are completely accountable for what goes out on their channels. A channel owner that sends out drivel or spam will not hold a large subscriber base for very long.

Thus, what we have is a full-blown, any-format, content broadcasting system in which anyone can own their own channel. Since "everyone has a printing press," distribution fans out exponentially among channel subscribers, and such a system scales in terms of bandwidth usage like a dream come true. Mark that as "anyone can own their own *unlimited bandwidth* channel."

2.3 Implementations percolate into user-space

What does channel-based sink discovery mean for end-users? First of all, it means no more searching. Instead, you find a few channels, subscribe to them, and relax while content arrives on the channels. After a while, you can evaluate channel owners' tastes and unsubscribe from the channels that disappoint you. Even if the channel owners are individually anonymous, you can build trust for them over time.

Playing the role of a passive sink in the distribution tree for someone else's channel will not satisfy most users for long, but this model makes it easy to become a content distributor. No one is forced or expected to play the role of the consumer. We can imagine a system in which nearly every node is both a source and a sink. Great responsibility is placed on the

shoulders of the channel owners with the largest subscriber bases—they might be at the top level of a multi-tiered, hierarchical quality filtration system.

2.4 Implementation-centric motivations

Much of our sink-discovery advocacy has centered around end-user issues, but our motivations for this model run right down into core implementation issues.

In the source-discovery model, for each piece of content distributed, every sink for that content contacts the content source directly. Thus, the source bears much of the resource burden (supplying the bandwidth needed to transfer to all of the sinks), while the sinks bear the brunt of the discovery burden (each carrying out the iterative search process until they find the content that they want). In the absurd source-discovery version of the author/book/publisher example, each reader must look the author up in the phone book and travel to the author's house, while the author must install a conveyor belt out of the front door just to meet the demand. This kind of *sink stampede* is common in most existing peer-to-peer systems: nodes holding fresh, popular content are swamped by the load of incoming sink requests.

For example, if a particular piece of content is to be distributed to 100 sinks using the source-discovery model, 100 users must go through the search process, and the source must provide bandwidth for 100 transmissions. Thus, 100 "units" of human effort are expended by the sinks, and 100 "units" of bandwidth are expended by the source.

In the sink-discovery model, both the resource burden and the discovery burden are spread evenly throughout the network. In terms of human effort expended for distribution of a piece of content, only one unit is expended: the source must discover the content and assess that it is worth distributing. In terms of bandwidth effort, the source and each sink only transmit the content to a handful of other nodes each. Likewise, in terms of network discovery, the source and each sink must discover a handful of nodes each (this is automatic and requires no human effort).

Overall, the total bandwidth and network discovery burdens are exactly the same, but they are amortized differently in the sink-discovery model. Specifically, the network bandwidth is shared among all content sinks, as is the discovery burden. More importantly, we must note that the amount of human effort expended to distribute a piece of content in the sink-discovery model is drastically less: in our example in which we distribute to 100 sinks, the human effort expended is 100 times smaller. Essentially, human effort expenditure is constant in the sink-discovery model, whereas it scales linearly with the number of sinks in the source-discovery model.

2.5 Weaknesses

What if the sources are not sending out what the sinks want? What if a particular user wants a piece of content that *none* of the channel owners are currently distributing? Our answer: that particular user should use a source-discovery system to obtain that piece of content. Indeed, source discovery has its place.

However, we claim that the need for source discovery is a fringe need—it is the exception, not the rule.

2.6 Intuitions about flocking

Our intuitions are as follows:

The majority of content exchanged in a system is the newest, most popular content. This fresh content uses most of the bandwidth, and most of the human effort is expended in finding it. In general, people flock to what is new, and their attention is short-lived.

Thus, there is a minor need to find and obtain yesterday's (or last year's) content, and source-discovery systems are perfectly suited to meet this need. When obtaining the latest content, however, a sink-discovery system is a more sensible choice. In fact, the latest, freshest content can be defined by the sink-discovery system itself, given that a quality filtration system can emerge.

3 Empirical results

To test the intuitions behind the sink-discovery model, we performed the following experiment on December 5, 2002.

3.1 A small experiment

Using a Mutella client [8] that we modified for our purposes, we connected to the Gnutella network and gathered all search queries that passed through our node over the course of 30 minutes. Next, we monitored search queries on the network over the course of 12 hours, split into half-hour intervals. During this monitoring phase, we tracked how many unique search words from the initial interval were repeated in each subsequent interval.

The initial half-hour interval had 569,200 search words, with 40,576 being unique. Over the course of the next 12 hours, the busiest half-hour interval had 915,741 search words, and the slowest half-hour interval had 548,962 search words. A total of 18,648,637 search words were seen during the experiment.

3.2 Major result

Figure 1 shows the fraction of unique search words from our initial half-hour interval that were repeated during subsequent intervals. To correct for the fact that some intervals had nearly twice as many search words as others, these fractions have been scaled by the number of search words that occurred in each interval.

This plot shows an obvious negative slope, indicating that particular search words in the Gnutella network have transient popularity. This result can also be interpreted as showing that flocking behavior is present in terms of what users are looking for in the Gnutella network. Even over the course of 12 hours, the population's interests can shift in a rather dramatic fashion.

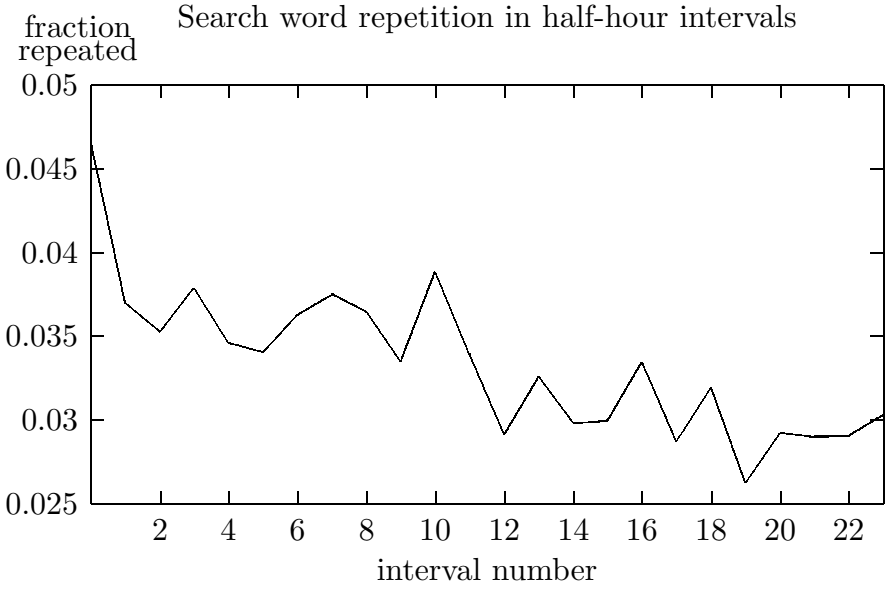

Figure 1: Search words were collected for an initial half-hour interval and then monitored over the course of the next 12 hours (for 24 additional half-hour intervals). For each monitored interval, this plot shows the fraction of the unique words from the first interval that were repeated at least once, scaled by the number of search words occurring in that monitored interval.

3.3 Minor result

In addition to making the previous measurement of search word repetition, we observed another interesting phenomenon.

What are the most popular search words used in the Gnutella network? Would this result be fit for publication? Those who would like to keep Gnutella's public image "family friendly" might sweat at the very idea of such a result. Our result, despite any cynical predictions, is clean.

Table 1 shows the top 10 search words used in the Gnutella network over the course of our 12 hour experiment. Our previous result showed that fine-grained flocking occurs, but this table shows that a coarse-grained pattern is also present: our experiment was performed in December, and "christmas" was one of the most popular search words. We conjecture that "christmas" would not be one of the top 10 search words at any other time of the year.

The other search words in the top 10 are not particularly noteworthy.

4 Proof of concept

A working sink-discovery peer-to-peer system, konspire2b, can be downloaded from:

http://konspire.sourceforge.net/konspire2b

frequency	word
82,613	mpg
94,394	i
100,364	christmas
126,685	in
127,139	a
127,851	and
129,239	avi
216,001	of
229,289	mp3
519,630	the

Table 1: The most popular search words in the Gnutella network over 12 hours.

References

[1] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.

[2] Gnutella. http://gnutella.wego.com.

[3] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *Workshop on Design Issues in Anonymity and Unobservability*, pp. 46–66, 2000.

[4] FastTrack. http://www.fasttrack.nu.

[5] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proceedings of the 2001 ACM SIGCOMM Conference*, pp. 149–160, 2001.

[6] edonkey. http://www.edonkey2000.com.

[7] B. Kantor and P. Lapsley, "Network news transfer protocol: A proposed standard for the stream-based transmission of news," *RFC 977*, 1986.

[8] Mutella. http://mutella.sourceforge.net.