CORANTE

from **Get Real** by **Marc Eisenstadt**

January 11, 2005

BitTorrent, eXeem, Meta-Torrent, Podcasting: "What? So What?"

SUMMARY: *The index that facilitates the sharing of files on a large scale is also the Achilles heel of peer-to-peer file-sharing, because it is vulnerable to litigation and closure. So what happens if the index is itself distributed? I try to get my head around the latest in peer-to-peer file sharing, and explain a bit about what I've learned, including the fact that BitTorrent's power rests in its 'swarm' distribution model, but not necessarily in your end-user download speed. What has this got to do with podcasting? (Answer: invisible P2P plumbing helps the podcasting wheel go round).*

[Warning: lengthy article follows].

First, some history

(skip ahead to the next section if you're already bored with the Napster, Gnutella, KaZaa, and BitTorrent saga).

Napster opened our eyes to the power of distributed file sharing on a massive scale. But it was closed down by lawsuits to stop it from listing copyrighted works for which the owners would naturally have preferred to collect royalties (there are thousands of commentaries on the pros and cons of such royalties, but that's not the focus of this posting). Successive generations of tools such as Gnutella, KaZaa, and now BitTorrent have created their own buzz, their own massive followings, their own headaches, and their own solutions to others' headaches. Here's my rundown of the 'big ideas' (and the people behind them):

Napster (Shawn Fanning): This was the Mother of big-time peer-to-peer (P2P) file transfers, i.e. my computer directly to yours, with a central server to maintain lists of who had what in order to initiate the transactions. It had a pretty decent user interface, plus the rapid growth, novelty, excitement and publicity that ensured plenty of good content. Those central server lists, leading to mass free trading of copyrighted material, also led it to be shut down.

Gnutella (Justin Frankel and Tom Pepper, creators of WinAmp): This was an open-source protocol that linked autonomous 'nodes' (users of the network) to other nodes, thereby eliminating the need for a central server list. Searching reliability varies, however, because it is subject to outages according to the connection/disconnection of individual users along the way. [**UPDATE** 13-Jan-05 - see NOTE 1 following the '*******' at the end of the article.]

KaZaa (Niklas Zennstrom and Janus Friis, who later created Skype): This technology built on a proprietary protocol called 'FastTrack', conceptually an extension to Gnutella, that deployed distributed 'supernode' search indices whose IP addresses were built in to the software, and which avoided the problems of (i) Napster's centralized lists and (ii) Gnutella's over-distributed nodes suffering outages and weakening the search. The prevalence of built-in 'adware' and the distribution of 'junk files' that masqueraded as originals were two of the weaknesses of the (still) wildly popular KaZaa.

BitTorrent (Bram Cohen): This was the next 'creative leap' in the P2P world, based on the following insight: distributing large files in fragments among large numbers of users, and requiring every downloader to be a partial uploader (of these fragments), enables the 'best of breed' of swarming behaviour -- as a file becomes more popular, so it becomes easier to download, rather than harder (as is the case with traditional file distribution)! A good overview explanation and a helpful analogy are provided in this excerpt from Brian Dessent's BitTorrent FAQ and Guide:

> BitTorrent is a protocol designed for transferring files. It is peer-to-peer in nature, as users connect to each other directly to send and receive portions of the file. However, there is a central server (called a tracker) which coordinates the action of all such peers. The tracker only manages connections, it does not have any knowledge of the contents of the files being distributed, and therefore a large number of users can be supported with relatively limited tracker bandwidth. The key philosophy of BitTorrent is that users should upload (transmit outbound) at the same time they are downloading (receiving inbound.) In this manner, network bandwidth is utilized as efficiently as possible. BitTorrent is designed to work better as the number of people interested in a certain file increases, in contrast to other file transfer protocols.

> One analogy to describe this process might be to visualize a group of people sitting at a table. Each person at the table can both talk and listen to any other person at the table. These people are each trying to get a complete copy of a book. Person A announces that he has pages 1-10, 23, 42-50, and 75. Persons C, D, and E are each missing some of those pages that A has, and so they coordinate such that A gives them each copies of the pages he has that they are missing. Person B then announces that she has pages 11-22, 31-37, and 63-70. Persons A, D, and E tell B they would like some of her pages, so she gives them copies of the pages that she has. The process continues around the table until everyone has announced what they have (and hence what they are missing.) The people at the table coordinate to swap parts of this book until everyone has everything. There is also another person at the table, who we'll call 'S'. This person has a complete copy of the book, and so doesn't need anything sent to him. He responds with pages that no one else in the group has. At first, when everyone has just arrived, they all must talk to him to get their first set of pages. However, the people are smart enough to not all get the same pages from him. After a short while they all have most of the book amongst themselves, even if no one person has the whole thing. In this manner, this one person can share a book that he has with many other people, without having to give a full copy to everyone that's interested. He can instead give out different parts to different people, and they will be able to share it amongst themselves. This person who we've referred to as 'S' is called a seed in the terminology of BitTorrent.

In the next section, I provide a little 'reality check' (showing why BitTorrent cannot deliver super-human download speeds as over-zealously implied in Wired and elsewhere), then talk about how the index information might itself be distributed around the net rather than hosted at key (vulnerable) sites, and what this has to do with podcasting, itself discussed in my earlier Get Real posting about it.

Interlude: A little reality check

BitTorrent is about super-swarms, not super-speeds. Here's why.

BitTorrent requires tracker sites to handle all the partial-fragment ~~negotiation~~ peer-peer introductions (think of the madness of the floor of the New York Stock Exchange, and you get an idea of the cool ~~juggling~~ mutual introductions that a tracker has to do). [**UPDATE** 13-Jan-05 - see NOTE 2 following the '*******' at the end of the article.] Fair enough -- after all, it's software, and it can cope. In fact, when you 'download a torrent', you are only downloading a small file (called something like 'video1.torrent') that itself has pointers to the tracker sites that handle all the traffic negotiation. There are 7 indirect levels involved, as follows: you follow a link (1) from some posting or listing site (2), to a torrent file (3), that you download, which, when loaded into the right application (4, identified further below), gives you a link (5) to trackers (6) that in turn manage pointers to the sites of all the fragments that need to be downloaded and stitched together (7), while being a good citizen and simultaneously dishing out a few fragments from your machine for good measure (and 'the right application' manages all this automagically). Still with me? In fact, you only notice the torrent file (item 3 above) and the site you got it from (item 2 above), and the rest happens without you needing to worry about it, given that you've got the right application (item 4 above), such as BitTorrent itself, or rivals such as burst!or any of the many others listed at places such as the excellent Wikipedia overview of BitTorrent, or the aforementioned BitTorrent FAQ and Guide.

But how fast does this all this file transfer magic actually happen? Remember that you're after 'the big download', which is all the stitched-together fragments (item 7 above) -- after all, you're trying to grab some enormous file, right? This big download takes at least as long as it would in the theoretical 'sole user' case, namely the case in which you were the only user on earth and had a dedicated connection at your maximum legitimate paid-for connection speed to the source of the file. Please re-read the previous sentence if you thought that BitTorrent would deliver you a 4GB Hollywood movie over a 1Mbps ADSL connection in 10 minutes. It won't. It is indeed awesome, but it's not a Time Machine. A 4GigaByte file, that is 4,000MegaBytes [in fact really 4096MB] = 34,359,738,368 bits to be precise (convert for yourself here) . Over a 1Megabit-per-second ADSL line (which is actually 1,048,576 bits per second), it would take 32,768 seconds, or 546.1 minutes, or 9.1 hours (not bad, in fact). That's the best case, if you had the pure and clean connection to the original source file all to yourself. BitTorrent is about clever swarming pools to spread the burden of distribution far and wide and thereby help maximize performance: it cannot get you a file faster than your connection speed can theoretically deliver!

All the BitTorrent gurus already know this, so what's the fuss? I've included this digression above because more than one authoritative source had me rubbing my eyes in disbelief and rushing off to get BitTorrent when I read about its superb download speed and general capabilities. For example, when Wired, January 2005, in 'The BitTorrent Effect' writes

> BitTorrent lets users quickly upload and download enormous amounts of data, files that are hundreds or thousands of times bigger than a single MP3.

you could be forgiven for projecting onto BitTorrent some Time Machine super-powers that may or may not be what the author intended. The author is generally more knowledgeable than me in these matters, so I figure perhaps he meant to say something like this:

"BitTorrent lets users pool their machines together and thereby eliminate bottlenecks on file transfers, ensuring that the payoff of such pooling grows significantly as files grow to sizes that are hundreds or thousands of times bigger than single MP3."

But hey, I ain't a journalist, and getting through my 'more-faithful-to-the-meaning' wording is a bit like getting through mollasses, so I can't entirely blame Wired! In any event, you can read reports, reviews, and blogs like this until you're blue in the face. To assess the actual performance gains, you need to go to the research literature where the definitive empirical studies have already been done. Enter Izal and colleagues to the rescue, with a great study entitled "Dissecting BitTorrent: Five Months in a Torrent's Lifetime"

Their article, worth reading if only for the detailed and no-nonsense description of how BitTorrent really works, puts this baby through its paces like no magazine or blog review you'll ever read. They studied server logs of *big* downloads (like the 1.77GB Red Hat Linux distribution) over a five-month period, involving some 180,000 clients. From their abstract:

> In this paper, we study BitTorrent, a new and already very popular peerto-peer application that allows distribution of very large contents to large set of hosts. Our analysis of BitTorrent is based on measurements collected on a five months long period that involved thousands of peers. We assess the performance of the algorithms used in BitTorrent through several metrics. Our conclusions indicate that BitTorrent is a realistic and inexpensive alternative to the classical server-based content distribution.

Bottom line? BitTorrent is for real. It achieves high performance in terms of 'throughput per client' and also in its ability to sustain large (~50,000) 'flashcrowds', i.e. the swarms that congregate during the early days of a popular download, and that with conventional server distribution would typically sink the server (or even a collection of mirror servers) of a large and popular download.

We wrote in "From BuddySpace to CitiTag" that "Big scale is an asset, rather than a liability." But we were just fantasizing – these guys (I mean the guys who dreamed up Napster, Gnutella, KaZaa and BitTorrent) make it true. But wait a minute, isn't there still a problem? Well, for 100% legal downloads, you now know all you need, so you can get hold of BitTorrent and check out the FAQ, use it to distribute your software, books, or music, and stop here, or skip ahead a few sections to get to the podcasting bit. Downloading of copyrighted material is illegal in many, if not most, places, so I strongly advise you not to engage in such practices. But you may be interested in the technology involved, purely as a thought exercise, or you may be interested in the right to distribute material without being subject to the scrutiny of authorities whom you feel are treating you or your companions unfairly, unwisely, or even illegaly. As did Thomas Paine in 1776, i.e. distributing his *Common Sense* pamphlet to hundreds of thousands of his cohorts in British Colonial America, thereby sowing the seeds of rebellion against the King of England. Or you may just enjoy reading about cool technology. Whatever. I advise you to stop reading, at once.

The next step: Meta-torrents

So far, so good. But all is not as rosy as it appears.

Anyone providing a site that does the job of 'item 2' in the 7-step chain listed above, i.e. the one that indexes, lists, or rates the quality torrents, is providing a rather valuable service, because BitTorrent itself has no built-in file searching capability. This makes BitTorrent well-suited for 'owner-idenitifiable' downloads, since people can list the relevant torrent files on their own sites. And this kind of 'tracker provenance' (saying where a file ultimately originates) is proving very useful for very large software distribution (such as Linux, for example). But for 'grey' downloads (either copyright-protected or in some other way not wishing the distributor to be widely known), any 'listing' service becomes so valuable that the leading 'quality torrent listers' such as suprnova are under intense legal scrutiny, and some of them are giving up to avoid the hassle.

This is where the next creative leap is required: What about using the very philosophy of BitTorrent, and indeed other P2P systems, for distributing the index listings, instead of having dedicated listing sites? Musing about this recently, I decided that such a technology ought to be called 'metatorrent': searching for this term on Google just the other day, I was astonished to see a mere 1 hit for the term! Had I thought of something for which the meme had not already spread? Alas, it was not to be. Searching for the pair of words "meta torrent" (or the hyphenated "meta-torrent") results in some more hits, so I was not alone – but wait, only 23 hits, in fact. Moreover, the first 10 hits I looked at used the term differently from me, and incorrectly in my opinion, to mean "a listing of torrent sites", just like the original-and-no-longer-listing-Suprnova and others, such as TorrentSpy. That's not what I meant by 'meta-torrent': to me, a true meta-torrent would use the very (torrent) technology and protocol to distribute a massive (and dynamic, and growing) set of torrent site listings. That's what makes it a meta-torrent, rather than a mere 'central/index site'. So my idea still seemed to have some original merit.

But how would a meta-torrent 'bootstrap' itself from no index at all to magically circulating itself among millions of machines? How the hell do I know... surely there must be 30 different methods for this! How do viruses spread? Trojans? Worms? Usenet postings? Social networks? Rumors? Flash mobs? Mexican waves? Spam? Memes? Blog postings? I have no idea! But I was certain that it would work. I imagined that once a meta-torrent was started (by any/all methods) it would self-perpetuate, change dynamically in the way that Usenet postings do (or used to do, in the days before dejanews and google groups centralised things to some extent), and acquire a life of its own. Listing sites like Suprnova were renowned for supplying quality torrents, but a meta-torrent would need some kind of rating or authentication system, no? Well, maybe it would self-regulate, or provide a mechanism for authenticated 'good sources', the same way BitTorrent uses hash coding to fingerprint all those hairy distributed fragments of a large file. Then something happened...

Enter eXeem

As I was reflecting on the meta-torrent idea just the other day, what should come to my attention but a BBC News Story, dated 7th January 2005, entitled File-swappers ready new network, about a new service called eXeem. It looked like the answer:

> Like BitTorrent, Exeem will have trackers that help point people toward the file they want.
> Like Kazaa these trackers will be held by everyone. There will be no centrally maintained list.

Holy smokes... this was obviously a big deal. I'm new to this, but it was evident that the big boys, who actually understood the technology, had hot implementations coming out the door. So now I could scrub 'meta-torrent' off of my New Year to-do list (actually this was short-hand for 'Hire geezer to work on meta-torrent', but yes, it really was on my list), and learn some more about it, which is precisely how this article came to be. I imagined that people must have been working on this like crazy for a long time. Sure enough, decentralized and anonymous file distribution are two of the holy grails of the internet, so I was right to cross it off my list.

But was eXeem for real? What did it do? How did it do it? Was it my meta-torrent fantasy come true? Some judicious Google and Technorati searching led me on a quick trail of postings on BoingBoing, Slashdot, and Mitosis, where the posters seemed to know what they were talking about (though even so, plenty of rumours and errors were being corrected in real time).

A guy called Simon posted the following beta test and screenshot info on Mitosis, including a nice intro with a user-friendly desciption of the BitTorrent technology, and why it needs things like a 'seed' file (the original complete file that kicks things off):

> *The problem:*
> All the info you need to have your bittorrent application connect to a tracker and start downloading is stored in a tiny text file with a ".torrent" extension. The problem has been where to get the .torrent files? There are a few websites that have created elaborate systems to offer torrents (as they're called) and display how many seeds and users are currently connected to that "torrent". Today, one of the largest of these sites, suprnova.org receives so much traffic that it has become a bottleneck in the system. Even worse, the dependancy on a website to get torrents has become a single point of failure.
>
> *The Solution:*
> What's needed is a program that decentralizes the way we find and distribute torrent and tracker data. The idea is to remove the single point of failure by having each person running a local application share torrent and tracker data with each other in almost the same way file data from a torrent download is shared with each other.

> Suprnova.org's vision of the future is called eXeem. It's an application that promises to change the face of P2P file distribution by encorporating bittorrent technology in a way that solves the problems listed above.

Simon goes on to provide tests, screenshots, and a brief review to the effect that he prefers to withold judgement until later releases. But it sounds promising.

A Slashdot article then makes a few related points about the relationship between Suprnova and eXeem:

> First, Exeem really isn't an extension of Suprnova as the hype might have you believe: the connection between the two seems more marketing than anything else.
> Second, , Exeem is pretty much what was rumored earlier: a blending of the tracker, the BitTorrent client, and decentralized indexing.
> Third, there's a mystery company. Someone is paying Sloncek. He won't say who, but there's a history in the p2p world of secretive development. Since Exeem is to be adware..

But this assertion that eXeem is adware is probably false, as far as I can tell, based on a couple of anonymous messages to BoingBoing, as follows:

> Following up on this previous BoingBoing post, reader Pseudonym says, in a rather hushed voice:
>
> Whois shows the crowd behind Exeem are in fact a company by the name Swarm Systems Inc. that are in fact located in Saint Kitts and Nevis, so would presumably be free from prosecution and lawsuits like Sharman Networks.
>
> And another anonymous reader (my, you're a sneaky lot) says,
>
> "Just wanted to let you guys know that exeem IS compatable with torrent files, you can load them up just like any other client. The ads sloncek was talking about are just ads not adware."

OK, I don't care much about the rumors, the ownership etc... I just want to know if eXeem implements the meta-torrent fantasy. Interestingly, the Slashdot article mentioned earlier also provides extensive comments on the pros and cons of similar meta-torrent ideas such as using Freeenet or Usenet to distribute the torrents. So the meta-torrent meme is certainly out there, in spades (if not in name). For example someone chimes in to the Slashdot thread with

> Anonymous bittorrent already exists:
> With all due respect to the Freenet team, they have done a lot of good work, but the network isn't designed for things like bittorrent. What you need is a low-latency network like TOR or i2p. With that said, anonymous Bittorrent already exists, its available to work on the i2p anonymous network. Just go to the i2p website, , install the software and then click on this: There are already bittorrent trackers on the i2p network. Why this hasn't been on slashdot is beyond me.

And in a related vein elsewhere, the man responsible for the only Google hit for *metatorrent* as a single word, l2oto (Roto Decker) on LiveJournal writes

> I'm architecting a system that combines email lists, gnupg, and bittorrent called sharemail. Signed torrent files attached to email lists makes for torrent subscriptions that resemble Konspire "push" p2p. That way, you can utilize mixmasters and other anonymous email mechanisms to protect the publisher.

BitTorrent creator Bram Cohen says in the Wired article that "The content distribution industry is going to evaporate... With BitTorrent, the cat's out of the bag" In a similar vein, it seems to me that the meta-torrent cat is also out of the bag. This is undergoing a lot of work right now, and if eXeem doesn't crack it, its successor surely will.

And what about Podcasting?

To close the loop on this discussion, consider podcasting as a time-shifted radio distribution model. In fact, podcasting generalises to RSS Media feeds, but let's just stick with podcasting, because it is simpler to understand. I summarised the 'so what?' of podcasting in an earlier Get Real posting, to the effect that it completes the 'last mile' of the connections from the user's point of view: you subscribe to an RSS feed that embeds within it (not unlike an email attachment) an MP3 file of interest to you, e.g. a regularly-scheduled technology review or talk radio interview, audio book, rock concert, etc., and presto-mundo, it appears on your iPod or other portable gadget whereupon you can listen while on the train, jogging, etc. All the pieces have been there for a long time, but podcasting makes it a hands-free seamless end-user experience (once you've done the one-time setup, at least), and that is extremely nifty. But there's still one piece missing.

There has been some concern expressed that RSS feeds (certainly full-text feeds) are themselves bringing the internet to its knees. This is probably something of an over-statement, but 'enclosures' could compound the problem. Consider this scenario: you have created a wildly successful weekly talk show, and the zillions of hits and downloads, whether directly or via RSS feeds, are killing your server, or forcing you to invest in mirror sites and similar server-centric distribution models. You are now 'a victim of your own success': large scale has proven self-defeating. But wait! The P2P visionaries rebel agains this very thought, remember? As I wrote above, "Big scale is an asset, rather than a liability". And in the BitTorrent world, massive scale improves throughput rather than thwarting it.

Sure enough, the guys behind podcasting are already way ahead on this one. iPodder, for example, is conducive to podcasters who make their MP3 RSS enclosures available as torrents. Setup is a little fiddly at this stage, but there are articles that provide how-to guides, such as "Battle the Podcast Bandwidth Beast with Bittorrent " Wahoo!! The loop is closed! There is end-to-end content creation and delivery for the masses, with no 'victim of its own success' bottlenecks. The more popular a file is, the more easily it can be distributed. Awesome.

That's the way the net was meant to be.

****

UPDATE NOTES 13-JAN-05

Though it can be confusing to change the wording of any blog entries that have already generated a lot of commentary, I thought it would be easier to introduce the two changes made above 'in-line' (using strikeout notation if a word has been deleted) and then explain them in the brief comments below, to preserve the context for all to see. A third note, not referenced above, is also added for good measure.

NOTE 1:
User ajs writes in this Slashdot posting that I got Gnutella a bit wrong:

> The network is quite robust and also possesses the multi-sourced download capabilities of BitTorrent. However, where BT requires a centralized "tracker", any node in the Gnutella universe can be a "tracker" at any time. This is the result of a protocol extension introduced quite some time ago (long enough that it seems to be widely supported by all of the clients that I connect to) where the client that you request a file from informs all of the nodes that it knows about who also have the file that they should contact you. They send you a UDP message indicating that they have the file, and you treat that much like a search result. Thus, when you search you might see 5 sources, but as you start to download, you immediately see that you're downloading from 50 hosts. It's pretty slick, and amazingly resilient.

NOTE 2:
Slashdot poster Burris complains about my overzealous use of the words 'negotiation' and 'juggling' as conveying the wrong idea about trackers:

> ... the tracker only introduces peers to each other. The tracker only knows which peers are finished and which aren't. Each peer then manages it's own "fragment-negotiation" which is really just downloading the rarest pieces from it's own point of view. There isn't any negotiation at all, really.

So I've edited the text accordingly to correctly stress the peer-peer introductions, which in my mind is still pretty frenetic, but Burris is probably right that I overstated the case, so I've softened the analogy.

NOTE 3:
In another vein, which didn't require any modifications, Fidgety Philip writes a Slashdot comment saying that my 'index = Achilles heel' opening summary point is biased, namely:

> The Achilles heel of *illegal* peer-to-peer, perhaps, but for those who want to share files legitimately, it's a strength, because it means that there is no need to blanket-ban the technology.

Good point -- legal uses of peer-to-peer don't necessitate some of the creative breakthroughs we are now witnessing, but I believe legal use will benefit from this creativity as well.

http://www.corante.com/getreal/archives/bittorrent_exeem_metatorrent_podcasting_what_so_what.php