<< Return to article

Pushing Peer-to-Peer

The networking approach that threatens to make the recording industry obsolete could also bring about a more reliable Internet.

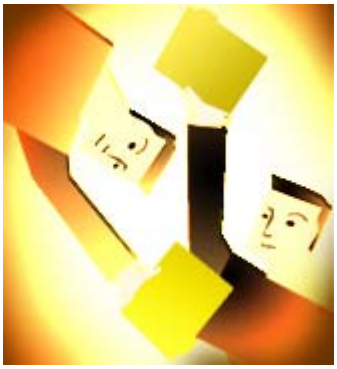

Illustration by Matthew Bouchard.

By Simson Garfinkel
The Net Effect
October 3, 2003

If I say peer-to-peer, you probably start thinking about those file-sharing services that let you get free music, movies, and pornography over the Internet. But peer-to-peer is about much more than violating the copyright of big record labels.

Indeed, although the term was coined just a few years ago, peer-to-peer is really how the Internet was originally designed to work. The theory was that all of the computers on the network would be first-class citizens, each capable of sharing resources or exchanging information with one another. Back then, a student at MIT might start typing on a computer in Cambridge and use it to log into a computer at Stanford. Meanwhile, another student at Stanford might use that same computer to log into the first system at MIT. Both computers would be simultaneously using and offering services to the network. The connections between them would be links between equals—that is, peer-to-peer.

As it turns out, most of the Internet didn't become a peer-to-peer system. Instead, the Net evolved along a different model. Low-cost computers, called clients, were distributed onto people's desks. These machines were used to access services offered by more expensive centralized computers—called, for lack of a better word, servers. Some of the earliest client-server systems let people share files by placing them on centralized file servers. The servers were also an ideal place to put electronic mail.

These days, the Internet's clients are the desktop and laptop computers that we are all so familiar with. The servers are the Web servers, mail servers, instant messaging servers, and other servers that our clients rely upon. There are even more servers operating behind the scenes—things like DNS servers that operate the Internet's Domain Name System, routing servers used for sending voice traffic through the Net (so-called voice-over-IP systems), and

even servers for companies that want to back up their computers over the network. The client-server model has been so successful because it's fairly easy to understand, set up, and maintain.

But there is a big problem with client-server architecture: it's vulnerable. When a single server goes down, all the clients that rely on it essentially go down with it. You can minimize this problem by having multiple servers, but then you have to make sure that they all stay synchronized. In fact, the server doesn't even have to go down—all you need is a break in the network.

Peer-to-peer is a fundamentally different way of thinking about the network—based not on the notions of clients and servers but on cooperation and collaboration. A peer-to-peer backup system might use all of the extra space on the hard drives throughout an organization to store extra copies of critical documents and personal e-mail; a peer-to-peer Web publishing system might use those same hard drives to store copies of Web sites. The theory here is that a thousand underpowered clients are still faster than the world's fastest server.

Unfortunately, peer-to-peer systems can be difficult to put together. The simplistic way to build one is to have each node report its presence to a central server. People who want to join the network then log in to the central machine use it to find their peers. While this works, it's not true peer-to-peer: shut down the central server, and the system collapses.

That's why most of the academic research on peer-to-peer systems has concentrated on building systems that work without any centralized control. This is harder stuff! Computers need to be able to discover new peers showing up, and be tolerant of peers that crash. Sometimes the network breaks into two or more pieces. Data needs to be stored in multiple locations. For an added challenge, try to handle potentially hostile peers that pretend to be good ones.

All of this experience could really pay off in ten or fifteen years. Consider these examples:

- One of the weakest points of the Internet right now is the domain name system, which is run by a loose confederation of name servers. Running DNS on top of a peer-to-peer system instead could dramatically improve its reliability.

- Today, if your business runs a small Web server and the site suddenly gets very popular, the server can crash from all of the extra traffic. But if all of the computers on the Internet were part of a global peer-to-peer Web cache, then small companies and individuals could publish their material to the multitudes. A good system would even prevent malicious modification of the Web page contents when they were served off other machines.
- In the event of a terrorist attack on the Internet's infrastructure, a peer-to-peer system would be far more likely to recover than a system that depended on top-down control.

Closely associated with the idea of peer-to-peer is the concept of an "overlay network." These are networks of computers that operate above the Internet, with direct links between computers that might be geographically distant on the Internet itself. Gnutella, Kasaa, and Morpheus are all overlay networks, as is the global network of Web servers operated by Akamai.

To understand why overlay networks are an interesting idea, you first need to understand how the Internet works without them. Normally, if a computer in Washington, DC, wants to open a connection to a computer in Tokyo, it simply sends the data packets into the vast soup that is the Internet. Eventually the packets end up on the other end.

Now, according to the original design of the Internet, the packets between the computer in Washington and the one in Tokyo would automatically travel along the fastest, most efficient path. Unfortunately, today's Internet doesn't work that way. Your Internet

service provider in Washington might have a sweetheart deal with an ISP in England to exchange packets over the Atlantic Ocean. The English ISP might, in turn, have a deal with an ISP in Germany. And the German ISP might have a special line that goes to Japan, but that line might be oversubscribed and slow. It could turn out that the fastest way to get packets from Washington to Tokyo is by sending them to San Francisco and then to Japan—a path that might exist but be discouraged by policy. This isn't a hypothetical example: such byzantine routing is common on today's Internet.

Overlay networks force the Internet to route packets differently by moving them between specific computers. For example, you might have an overlay network that consists of a computer in Washington, another in San Francisco, and another in Tokyo. By sending the packets from one of your computers to the next, you could defy your ISP's routing policy, and force your packets to go along a path of your choosing.

If you have access to computers at more than one location, you might have experienced the wonders of overlay networks yourself. For example, I have a computer in Belmont, MA, another in Cambridge, and another in Boston. Each machine is served by a different ISP. Sometimes I can't open a connection between the computer in Belmont and the one in Boston. During these times, I can hop from Belmont to Cambridge, then from Cambridge to Boston. Essentially, I've created my own overlay network.

Most peer-to-peer systems create overlay networks on the fly whenever they need to overcome congestion or routing problems on the underlying Internet. Overlay networks are also a great place for academics to experiment with new routing algorithms—algorithms too new and untested to let them loose on the Internet infrastructure. Peer-to-peer is pretty powerful stuff. What we've seen so far—the copyright infringement systems—is really just the beginning. Peer-to-peer could overcome many of the fundamental problems that are facing the Internet today—problems of centralized control, vulnerable servers, and the difficulty that most organizations have scaling. On the other hand, peer-to-peer could also make the Internet's security problems worse, by allowing hackers to create large-scale attack networks. Peer-to-peer could be a boon for the artists and the recording industry, giving them a way of publicizing and distributing their intellectual property for far less than they do now. Yet better peer-to-peer systems could further hurt the recording companies—and not just through copyright violations.

Already, today's peer-to-peer networks do a better job distributing music than the labels do; next-generation networks could implement systems for promotion and even collaborative filtering to make it more efficient for users to find the music that they want to hear. Peer-to-peer systems could even act as a kind of Internet radio system, eliminating the need for radio play and the accompanying payola. This was fundamentally Napster's plan, as the record labels learned during the process of discovery during their lawsuit. The real threat that peer-to-peer poses to the record labels is that it could make them obsolete.

At the end of the day, peer-to-peer technology is about increasing the reliability and the redundancy of Internet-based systems. That's why the recording industry is afraid of it—because peer-to-peer can be used to create networks that the industry can't shut down. But peer-to-peer can also be used to create networks that earthquakes, wars, and terrorists can't shut down. Ultimately, I think that we're better off trying to strengthen the Internet rather than trying to make it weaker.

Simson Garfinkel is an incurable gadgeteer, an entrepreneur, and the author of 12 books on information technology and its impact.