P2P For Communications: Beyond File Sharing

Cullen Jennings and David A. Bryan

Dr. Cullen Jennings is a Distinguished Engineer with Cisco Systems specializing in VOIP, SIP and security. He is responsible for standards, forward looking architecture, and technology for the Voice CTO office. He can be reached at 408/421-9990 or fluffy@cisco.com. David A. Bryan is the CTO and founder of SIPeerior Technologies (www.sipeerior.com), targeting the P2P SIP market. Mr. Bryan is actively involved with the IETF and open source telephony projects. He can be reached at dbryan@sipeerior.com.

P2P can be used for more than free phone calls and illegal file sharing.

Peer-to-peer (P2P) telephony and communications applications are generating tremendous interest and concern in the telecommunications community. EBay's acquisition of Skype underscores the growing attention vendors are paying this technology, while BitTorrent and other file sharing programs threaten to taint the whole idea of P2P with connotations of lawbreaking and copyright infringement.

Vendors want to learn how P2P might help or hinder their IP-telephony products. Network operators wonder how they can keep P2P from overrunning other flows and disrupting their traffic management expectations. Meanwhile, industry researchers are exploring ways that P2P can be put to work in various network scenarios.

This article aims to explain what P2P is all about, and put some recent implementations and possible uses for P2P into perspective. Although the authors are actively engaged in P2P and SIP research, we have tried to address the industry at a general level and from a non-biased position.

What Is P2P?

Defining P2P technology isn't really difficult, although many current definitions have more to do with marketing than with technology. At the most fundamental level, P2P simply implies that multiple pieces of software work together directly as "peers," rather than as "clients" that need a centralized server to help process information.

From an architectural standpoint, P2P is just about the opposite of client/server, although P2P applications can be indistinguishable from more traditional applications. It is more accurate to view P2P as a philosophy of reducing or eliminating the need for central servers, rather than as a particular, clearly defined architectural approach.

These days, the term P2P is most often used in reference to end-user applications that communicate with one another or collaborate among themselves with little or no involvement from centralized servers. Music and movie file sharing is probably the most infamous example of these applications, although some of these systems make use of central servers.

For example, hybrid P2P systems such as Skype, Napster and even traditional SIP use a central server to locate the other participating clients and P2P mechanisms to exchange files or media. In contrast, pure P2P systems like Chord distribute both the look-up mechanism and the media exchange among the peers.

In pure P2P, two techniques are available to locate files and other resources:

- **Unstructured P2P**, which uses techniques such as "flood" to locate resources. Flooding involves asking all nearby nodes, and having them ask their neighbors, until a result is found or until no result is found and the search parameters have been fulfilled, indicating the user is not in the system.
- **Structured P2P** uses a mathematical technique in order to determine where to store and subsequently find information. The structured method produces more efficient searches, at the expense of adding complexity.

Over the years, P2P has been used for many purposes other than file sharing. Several well-established Internet mechanisms—including the Simple Mail Transfer Protocol (SMTP) for the exchange of email between various servers and the Border Gateway Protocol (BGP) mechanisms used between routers to update routing information—are arguably P2P, as instances interact with each other as equals.

Other P2P examples include Apple's open-sourced, enhanced version of TCP/IP, the Rendezvous protocol (which requires no Dynamic Host Configuration Protocol, or DHCP, server), and the IETF's service location (SRVLoc) protocol for announcing and discovering network services. Both eliminate the need for central servers and have been used constructively for years.

Like most technologies, P2P can be used for positive and negative applications, and the technology should not be ignored because of a few inappropriate uses. That said, negative connotations have a tendency to stick, and the term "P2P" is already at risk of joining disparaged terms like "hacker," which used to mean a skilled software engineer adept at quickly adapting code. It will be interesting to see if P2P becomes a tainted term.

Registration And Resource Location With P2P

There are a number of places in a communications system where P2P technology could be employed successfully. One of the most intensely discussed in the IETF's informal P2P SIP meetings, is to use P2P technology for distributed registration and resource location.

Registration is the mapping of a persistent "user name"—such as a phone number, SIP URL or Skype username—to the routable (IP) address where that user currently can be reached. This address can change frequently, as users switch among home phones, mobile devices, office phones and softphones. Despite these different devices (or resources) with their different routable addresses, the user will always want to be reachable using the same "user name."

In P2P systems, algorithms determine how the users' registration information is distributed, searched for and retrieved by the peers

Registration and resource location can be accomplished through several mechanisms, including centralized servers, network broadcast or by pre-provisioning devices with a list of relevant identifiers and addresses for other devices.

In a traditional client-server communications system, a central server stores all the users' registration information. Each client device is pre-provisioned with the address of the central server. The client registers with the central server, and its future requests to contact other users also are handled by the central server, which forwards these requests to the destination users' currently registered devices.

In a P2P system, an algorithm—typically, a distributed hash table (DHT) algorithm—determines how the registration information is distributed, searched for and retrieved by the peers. This algorithm is pre-provisioned on the peers as part of the P2P application.

Peers locate one another via broadcast or pre-provisioning mechanisms, and each participating peer agrees to store some registration information about other peers. Registration information is usually replicated on multiple peers for reliability.

For example, P2P user Bob wants to locate P2P user Alice, so Bob's device queries neighboring peers to locate one that stores Alice's registration. Then Bob's device can use that registration information to contact Alice directly at her network address. (If Bob already knew Alice's address, he could use that to contact her directly.)

Another application of P2P technology anticipates using the same functionality to locate distributed media relays and other functions ordinarily provided by central servers. For example, some Network Address Translators (NATs) and firewalls require that devices inside different NATs communicate with one another only by sending their messages through a relay server that is outside of both of the NATs (see *BCR*, January 2006, pp. 15–16). Using P2P, a device behind such as a NAT could query its peers to locate a node that can be used as a media relay, has available bandwidth and is located outside the NATs.

Systems based on this technology are being actively studied and developed, although some challenging problems remain. The most difficult is the "distributed identity" problem: What prevents multiple users from claiming to be the same person? How can users be certain that the person registered as Bob is the same person each time? Without a central server, how can a peer verify Alice's identity? This is still an open issue in completely distributed registration systems.

Sharing Storage And Building Ad Hoc Networks With P2P Technology

Another P2P application, distributed storage, anticipates additional (and more legitimate) uses for online file sharing. Instead of swapping entertainment files (or parts of files), peers could store encrypted backup versions of important files for one another—such as buddy lists, voice mail messages and device configuration information.

Since these files are encrypted, the data could be securely stored, even if its exact location is not known. Then participating peers can migrate easily from device to device, without having to manually update configuration files, move address books or the like. Of course, such scenarios also require standardized buddy lists and address books.

Peers could also be used to extend the reach of wireless communications devices. If an emergency call needs to be routed in a wireless network, but that device has no access to a base station, the traffic could be relayed using other peers.

Turning to products—rather than components of existing products—that could be leveraged with this technology, researchers see a number of opportunities:

- Serverless PBX applications for small offices seem to be currently receiving the most attention. These systems, such as Nimcat (recently acquired by Avaya) and Peerio (from Popular Telephony), have a collection of nodes that are, in general, geographically close, and that communicate with each other directly, as well as storing resources and registrations for each other. New nodes can join simply by entering their name or number and require little or no configuration. Similarly, gateways that participate in the same configuration can be used to handle PSTN lines and redirect these to the nodes. P2P communications can also be used as a backup to client-server communications, perhaps during server maintenance or outages.

Skype has taken a proprietary approach, while GoogleTalk is more standards-based

- Emergency services is another hot area where P2P technology may flourish. To set up a wireless network, a group of first responders could simply power up wireless access points and devices, and the nodes would locate one another, exchange information and self-organize (see *BCR*, January 2006, pp. 61–65). New units called in to participate can join the network without any pre-provisioning, since the group of nodes stores the necessary information. Broadcast-based techniques can also be incorporated to allow for information to be widely dispersed.
- Large scale, global communications systems such as Skype are another attractive application (more about Skype and other existing P2P products below). Eliminating or reducing the number of servers eliminates as much central control as possible from these systems. This has a number of consequences, including lower operational costs and lower risk of failure. Essentially, unless Internet connectivity is interrupted, it is difficult for such systems to fail completely. If a device can reach one peer, it is connected to the network, even if a large number of other peers fall off the network.
- Distributed storage and replication of files, as mentioned above, is another attractive use. For mobile devices, set-top boxes, or personal entertainment devices, the possibility of obtaining files (such as ring tones, programming or other media) from other nearby devices could be a boon, not only to the customer but also to the service provider that wants to reduce network traffic and limit loads on centralized file servers.

All these technologies will be far more likely to succeed, and far easier to develop, if they are based on widely deployed and accepted Internet standards, such as those being discussed inside IETF. The industry has made a large investment in standards-based technology, including handsets, gateways, and even devices such as SIP-aware firewalls. If P2P technology can be tied to these existing standards, the large investment, in terms of both purchased equipment and the developed technology components of these devices (such as software stacks) will be able to be reused.

On the other hand, if closed, proprietary “standards” dominate this space, existing technology is much less likely to be reused, and companies developing or deploying solutions in this space will face vendor lock-in. One of the original promises of SIP and other standards-based VOIP technologies was an end to vendor lock-in, and the authors hope this ideal may also flourish in the P2P VOIP space.

Existing P2P Products

While P2P research is ongoing, there are a number of popular deployments and efforts in the field. The SIP protocol is perhaps the most obvious. While not entirely P2P, SIP was designed from the outset to be peer to peer, once the user’s routable IP address has been discovered.

In fact, the main need for a central server in SIP is to locate other users. If you already know another user’s IP address, or have some other way to find it, you and the other user can communicate using two off-the-shelf SIP devices without a central server. SIP also transmits content and implements a large number of features without the use of a central server.

While the Skype protocol is closed and publicly-available details are thin, research indicates that Skype works in somewhat the same way as SIP systems, with some critical differences:

- Like SIP, Skype uses a central server to authenticate and verify that users are allowed to use the system, but Skype uses a proprietary protocol for this service.
- Unlike SIP, each Skype node may be used as a media relay to assist with NAT traversal. This, along with a very effective and streamlined configuration process, is part of what allows Skype to work so smoothly.

If a media relay is required for NAT traversal, Skype easily finds one within the system. This reduces the load on Skype’s own servers, which only need to be used for authentication and lookup. The downside, at least from a network administrator’s point of view, is the unpredictable traffic load on Skype nodes within the network that are functioning as media gateways.

Since any node meeting a set of requirements identified by Skype (presumably, having available bandwidth and a public address) can easily be turned into a media relay for many other calls, participating in a Skype network leaves the user open to being used as a Skype relay, with the associated bandwidth problems, whether the user’s network administrator desires it or not.

Similarly, a user’s calls may be routed through some random location during a Skype call, although the encryption used on Skype calls makes this a less serious concern.

With these two exceptions—using nodes as media relays and the proprietary protocols—Skype is very similar in end user experience to other solutions, such as Google’s new Google Talk service, although Skype has taken a non-standard approach, while GoogleTalk is largely standards-based.

Extensive non-commercial efforts have been focused on extending SIP to use a pure P2P-based mechanism to locate other users, leading to drafts and papers being published by the authors; by researchers at Columbia University and The College of William and Mary; and by others. P2P SIP has drawn healthy participation on SIP-related IETF mailing lists and in BOFs at the last three IETF meetings. (For more information, readers are referred to www.p2psip.org, which also provides links to research, IETF efforts and background reading on P2P SIP.)

Several companies, including Avaya (via Nimcat), Popular Telephony's Peerio group, Cisco's Linksys One and others, have been developing proprietary P2P- and SIP-based approaches to the challenge of creating small business phone systems with minimal configuration or administration requirements. Finally, EarthLink has developed file sharing software that uses SIP rather than P2P to locate other users, essentially reversing many of the systems described above. Their work uses SIP messages, as well as existing IETF standards for NAT traversal and security, to build a file sharing system.

The Road Ahead (And The Speed Bumps On It)

While P2P is a very exciting area, it is not without challenges. Security, in particular, is both important and difficult. Some of the same mechanisms that give P2P its scalability and extremely simple configuration properties also make it hard to secure. A truly distributed *and* secure P2P system remains the subject of active research and is likely to be a difficult problem for some time.

For example, with limited or no centralized control, there is always the risk that nodes serving as relays might intercept calls (although end-to-end encryption can prevent this), that users might spoof one another, and that users might hijack other users' calls.

Research in this area is still ongoing, although for many applications in which P2P is proposed—*ad hoc* communications and small office deployments, for example—these risks may be an acceptable trade off for the scalability and ease of configuration that P2P communications offers. Similarly, further work is required in completely decentralized systems to eliminate the "distributed identity" problem described above.

Using a central server to authorize and validate users is an attractive stop-gap remedy taken by applications such as Skype. The secure email approach—that is, to have peers use centrally-issued certificates—is also much discussed. While this also introduces a central server, it is only consulted the first time two peers communicate. The "web of trust" model has also been proposed—in which you trust your friends, and assume that others whom your friends trust are good as well—to determine which nodes are allowed to participate in the network.

Emergency calls are also an important unresolved issue. Many current VOIP 911 solutions involve carefully pre-provisioning each device with information about its physical location and where to call in the event of an emergency. In a distributed system, this can be very difficult.

Lawful intercept is similarly difficult, since there are no predictable central servers through which information can flow, and no responsible party who can be ordered to intercept the information. The difficulty of lawful intercept is the flip side of the difficulty in censoring a network—a network that is fully decentralized is resilient to attacks to shut it down, but is also very difficult to intercept, particularly if encrypted.

The same mechanisms that make P2P scalable and simple also make it hard to secure

Various approaches to implementing P2P for communications in a standards-based way are also being discussed in the IETF, and the most suitable have not yet been determined. Should an already established P2P algorithm (such as Chord) be used, or should new algorithms be developed? Should the messages needed for P2P be sent using current protocols, such as SIP, or should new protocols be developed?

P2P communication is at an exciting stage of development. The technology is mature enough to be deployed in a hybrid way, and to be debated among the standards bodies, and yet is still a rich area of research with many open questions.

P2P also poses a number of challenges to a network or system administrator:

- How does one differentiate legitimate traffic from inappropriate file sharing?
- How does one block inappropriate file sharing, and prevent it from being tunneled over communications channels?
- How should important P2P voice traffic be prioritized?
- For proposals involving peers acting as media relays, how do you ensure that your organization isn't serving as a relay for other users'—perhaps even competitors'—traffic? Worse yet, how do you prevent your voice traffic from being relayed through a competitor?

While encryption, traffic shaping and firewalls can address some of these issues, many are more in need of policy than technology solutions. The implications for managers are murky, and weighing the advantages of P2P communications against the potential risks is likely to be an issue for IT departments for some time to come.

Conclusion

Perhaps the greatest potential of peer-to-peer software is twofold: To make systems in which the capacity of the system scales up at the same rate that new users join the system; and to create systems that are easier to deploy and administer. The greatest limitations of P2P systems revolve around the difficulties associated with security and applying centralized policy.

P2P technology probably will prove very useful for certain classes of communications, but it won't be a silver bullet, nor will it completely change communications. Many existing

Vendors and customers are likely to use both client/server and P2P systems

centralized servers will be far better equipped to meet certain deployment needs than P2P solutions, and these technologies are likely to co-exist as pieces of an organization's communications solution.

End users won't know if an application is purely P2P, hybrid P2P, or fully centralized, and most are unlikely to care. Skype, Yahoo!, and Google Talk can give very similar functionality with radically different technology, but the consumer's choice, as always, comes down more to end user experience and price than to technology.

Many deployments will likely take advantage of P2P for some aspects of their solutions while leaving central servers in place for other features. The balance between client-server and P2P solutions will be yet another component in designing communications networks, and like most systems, administrators and vendors are likely to build their solutions by combining both technologies in a way that makes the best sense for their particular deployment or market.

Some P2P software is being developed in ways that allow users to circumvent the desires of firewall and security administrators. As the technology matures, firewall vendors are addressing these concerns by incorporating technology to allow control of known P2P applications. While some P2P applications have no desire to circumvent administrators, those that do will cause a constant iteration between application developers eager to have their traffic pass, and firewall vendors seeking to enable administrators to retain control of their networks.

If, in the end, P2P technology turns out to be a good thing and is viewed as a benevolent new technology, many existing applications will—in many cases rightly so—be rebranded as P2P. On the other hand, if the name P2P becomes, or already has become, too tainted by illegal file sharing, the technology may still be deployed, but it won't be called P2P.

IT departments should not be afraid to deploy P2P systems that reduce work for the administrator and end user, but should think carefully about the policies they need to enforce and how that will be done in a P2P environment with distributed data and processing□

Companies Mentioned In This Article

Avaya (www.avaya.com)
Cisco Systems (www.cisco.com)
EarthLink (www.earthlink.net)
Google (www.google.com)
Popular Telephony (www.peerio.com)
SIPeerior Technologies (www.sipeerior.com)
Skype Technologies (www.skype.com)