Disruptive Telephony

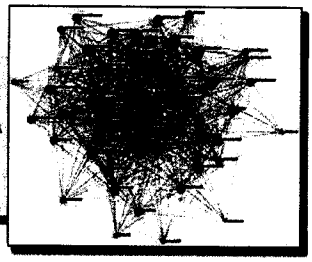
Dan York on how Voice over IP is rewriting (almost) everything you thought you understood about telephony...

Disruptive Telephony
Disruptive Conversations
Code.DanYork.com
Twitter: @danyork
Voice of VOIPSA
Blue Box Podcast
DanYork.Me

November 01, 2010

A Brief Primer on the Tech Behind Skype, P2PSIP and P2P Networks

What is an overlay network? What's a DHT? How does a node compare to a supernode? What differentiates a "pure" peer-to-peer (P2P) system from a "hybrid" system?



UPDATE: Unfortunately, this post is no longer accurate with regard to Skype's infrastructure. After the massive Skype outage in December 2010, it was expected that Skype was exploring ways to make their system more stable and resilient. In early 2012, Skype (at that point now owned by Microsoft) was reported to have replaced much of the P2P supernode infrastructure with supernodes hosted in Microsoft data centers. Since that time we've understood that additional changes have been made for resiliency's sake. Given all that, it's hard to know exactly how Skype's infrastructure exists today. This article below does, though, provide some background into the basics of peer-to-peer (P2P) infrastructure.

I have a series of posts planned over the next few weeks related primarily to Skype and some of the changes brought about in Skype 5.0 for Windows that are interesting from a technology point-of-view - but in order to write those posts, I need to build a bit of a foundation of some of the issues and terminology used in peer-to-peer (p2p) networks.

If you are *not* familiar with peer-to-peer (p2p) networks and the terminology associated with them, my goal is to give you a basic background. If you *are* familiar with P2P technology, well, you can probably skip this and wait for the next post :-)

Peer-to-peer vs. Client/Server Networks

For starters, let's differentiate between "*peer-to-peer networks*" (p2p) and more traditional networks. If you think about running, say, AOL Instant Messenger (AIM), on your computer, you install the AIM *client* software on your computer. When you launch the software, it connects you to the AIM *servers* running somewhere out on the Internet. When you send a message to another AIM user, that message goes from your client up to the AIM servers and then from there down to the recipient's client.

This is "*client/server*" networking and is how the majority of the Internet-based services we use are structured. We operate web servers, mail servers, file servers, etc., etc. and have local clients installed on our computers that connect to those servers. Even in many of the "cloud" services out there, we are still using a client (typically a web browser) to access services running on some big set of servers out on the Internet. (And the services may in fact be client/server-based, but with all of that being hidden in internal infrastructure.)

Peer-to-peer networks are completely different in that:

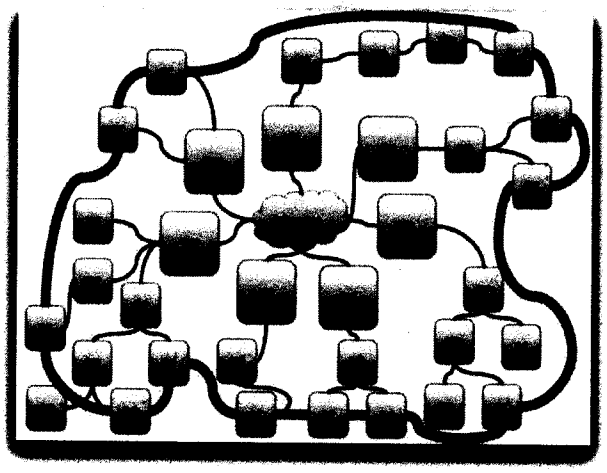
> *There are NO servers.*

(Well... at least in "pure" P2P networks. We'll talk about hybrids later.)

Instead, every "node" on the network is a "peer" - it is ***both*** a client *and* a server. It acts as a client accessing resources from other peers - and acts as a server providing resources to other peers. The computers that are running the p2p software are all participating in a "p2p network" connecting the computers (nodes) together.

Overlay Network

While I've heard people lately talking about that p2p network as a "p2p cloud", the technical term is to refer to it as an "overlay network" (or "network overlay" or just "overlay"). The idea is pretty straightforward... you are running software on your computer that "joins" a network that exists *on top of* the actual Internet connections. Here's a picture of how it might look:



The "Internet" is in fact a collection of top level Internet Service Providers ("ISPs") who run the "backbone" of the Internet and connect out to 2nd tier ISPs who connect to 3rd tier ISPs who connect to... etc., etc. until you actually get down to the network drop or wireless service you use to get access for you PC (using "PC" generically here for a computer... could be Windows, Mac, Linux, whatever). Everything is all interconnected via IP addresses and running the TCP/IP protocol suite at the networking layer.

The "overlay network" is created on top of that IP network by PCs running the P2P software. They connect to each other and communicate with each other regardless of what the underlying network infrastructure actually looks like.

Now, I've shown it in the drawing as a nice somewhat circular ring because it was convenient to do so. In reality the picture would undoubtedly be a whole lot messier - but ***logically*** it would still function a lot like a ring in most P2P networks. (And note that some people do talk about an overlay networking being a "p2p ring".)

Distributed Hash Tables (DHTs)

A question, of course, is ... *how do those peers know where each other are?* If the overlay network is on top of the IP network... still, ultimately the data connections have to happen over IP - how does a peer know where to send packets?

Now ***this*** topic gets into a whole area of research that network geeks like me may find absolutely *fascinating*... but everyone else may find their eyes glazing over.

Suffice it to say that there are a whole number of different algorithms for building p2p networks, and most today use some form of a *distributed hash table (DHT)*. The idea is that you have some piece of information, like my user name, for which you want to look up other information such as how to reach me. Your P2P software performs a "hash" on that info (called the "key") and winds up at an address for that information *inside the overlay network*. It then queries the overlay network and the node responsible for that "address" responds back with the requested information (the "value").

Think of a DHT as a giant distributed database of "key / value pairs". (And in fact some large distributed database systems use DHTs.)

There is a *LOT* more I could get into... but that would quickly take this out of "A Basic Primer". Those wanting more info should check out the DHT entries on Wikipedia and the P2P Wiki, both of which are good. I mention DHTs here primarily because any time you go looking for P2P info on the net, you pretty soon wind up reading about DHTs.

Service/Resource Discovery

If every peer can act as a *server*, how do other peers know what services a given peer can offer? In our real-time communication world, one peer might have a PSTN gateway connected to it. Or maybe an automated attendant... or a voicemail server. Or maybe it's running on a mobile client with reduced capabilities.

One challenge in a P2P network is to learn what resources each node has. This is typically referred to as "resource discovery" and again, different P2P networks have different methods of solving this issue.

Directory

Related to resource discovery is the whole "directory" issue - how do you find other *users*? If you think of Skype - or any P2P communication system - if I want to call or IM someone, how do I know where they are in the P2P network?

Enrollment / Authorization / Authentication

There are two security-related issues that come up (well, there are *many* but two key ones). One is - *how does a new node join the p2p network?* And the second is - *how does an existing node that dropped out of the p2p network (perhaps going offline) get authorized to **re-join** the network?*

Some P2P systems refer to this first issue as one of "enrollment." How does a node "enroll" in the p2p overlay network?

The second question is one of authorization / authentication of the node that it is allowed to join the p2p network.

Again, there are multiple ways to solve this issue including, at least in Skype's case, of adding in some *servers* to the mix to perform these functions.

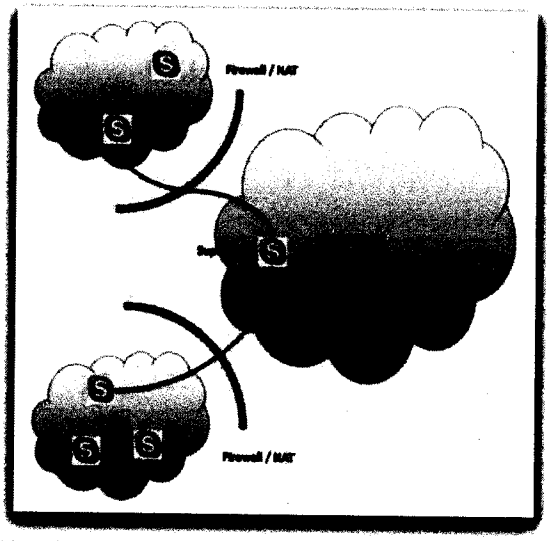
Churn

A fundamental challenge of any P2P network is how to deal with the "churn" of nodes dropping in and out of the overlay network. For instance, as computers go offline and then come back online, they leave the overlay network and then rejoin. Different overlay network topologies have different strategies for coping with churn and minimizing its impact.

Nodes vs "Supernodes"

A second fundamental challenge for P2P networks is our dear old friend "Network Address Translation" (NAT) and all the related firewall fun. Given that a node behind a firewall *cannot **directly** communicate* with a node behind another firewall, how do they get connected?

The idea is that you have another network node that is ***not*** behind a firewall that can act as a relay between the two nodes behind firewalls. In Skype's terminology, these relay nodes are referred to as "supernodes".



UPDATE: Shortly after I published this post, several friends suggested that I was making this perhaps *too* basic with regard to Skype. In fact, several deep and long posts could easily be written about Skype's supernode architecture as it is quite fascinating.

The point I was trying to make was that for nodes behind NAT/firewalls to communicate with other nodes behind NAT/firewalls, there needs to be some node *outside* of firewalls - on the public Internet - that can broker the communication between the endpoints.

In the case of Skype, we typically refer generically to those public nodes as "supernodes". In fact, the nodes known as "supernodes" perform the somewhat limited functions of connecting nodes together, providing a distributed database and choosing appropriate nodes to act as "relay nodes" when necessary.

These "relay nodes" are, in turn, the ones that perform the actual relaying of calls, messages, packets when direct connection is not possible. It *is* possible that these relay nodes could be located behind a NAT/firewall as the supernodes are connected to them and using them essentially to offload processing. Skype provides more info (including how to specify that your systems *not* be used as supernodes) in the IT Administrator's Guide I link to in the resources below.

In the IETF world of SIP, this external connection function is performed by either STUN or TURN servers (and that, too, could take up several blog posts).

"Pure" vs Hybrid P2P Networks

In an ideal world, a "pure" P2P network *has no servers whatsoever*. Every node on the network is a peer and functions as both a client and a server.

Sometimes, though, reality intrudes. Sometimes servers may need to be introduced into the mix in order to provide specific services for performance, security or other reasons. For instance, in Skype's case the general understanding is that servers are used for enrollment (to have a Skype client join the P2P overlay network for the first time), for some degree of authentication and also for connections out to the PSTN.

Networks that introduce some level of servers are typically called "hybrid" p2p networks.

That should be enough...

... for me to get started with my articles. I hope this was helpful - and if you have questions about items here or additional points you think I should mention, please feel free to leave them as comments.

Resources

Lots of good info out there about P2P networks in general... here are some you may find useful:

- Wikipedia:
  - Peer-to-peer
  - Overlay network
  - Distributed hash table
  - Skype
- P2P Wiki (and its article on DHTs)
- P2PSIP.org - lots of good info here
- IETF:
  - P2PSIP Working Group
  - Concepts and Terminology for P2PSIP - good draft to understand IETF's terminology
  - P2PSIP Base Protocol Internet-Draft
  - For those looking for a background in P2P security issues, I assisted with a now-expired Internet Draft on P2PSIP security issues which you may find interesting.
- Skype: P2P Telephony Explained
- Skype: IT Administrator's Guide to Skype (PDF)

If you have other suggestions, please feel free to send them along.

Flickr credit: porternovelli - and yes, this image really has nothing to do with P2P networks - it's a graph of top PR Twitterers - but it could be for a P2P network... and in a way all of those Twitterers are an overlay network ;-)

If you found this post interesting or useful, please consider either:

- *subscribing to my email newsletter;*
- *following me on Twitter; or*
- *subscribing to the RSS feed*

Posted by Dan York on November 01, 2010 in P2P, SIP, Skype | Permalink

TrackBack

TrackBack URL for this entry:
http://www.typepad.com/services/trackback/6a00d8341bfc6e53ef0133f5817b6797ob
Listed below are links to weblogs that reference A Brief Primer on the Tech Behind Skype, P2PSIP and P2P Networks:

» Skype and the Incredible Power of Persistent Group Chats from Disruptive Telephony
What is one reason why many people continue using Skype for chat / instant messaging when so many other solutions are out there? Particularly when Skype chat is a closed, proprietary "walled garden" that doesn't interact with IM networks? After... [Read More]

Tracked on December 03, 2010 at 12:10 PM

» Understanding Today's Skype Outage: Explaining Supernodes from Disruptive Telephony
For the first time in 3 years, Skype was down today - and as I write this is still in the process of slowly coming back online. A ton of articles were written today, mostly all pointing back to Skype's... [Read More]

Tracked on December 23, 2010 at 03:26 AM

» Skype to boost headcount by 50% this year and offer SLAs from Disruptive Telephony
The Financial Times is out this morning with an article about Skype CEO Tony Bates and his plan to hire around 400 more people this year. The article offers some insight into his thinking, and included this piece related to... [Read More]

Tracked on January 17, 2011 at 10:11 AM

» The End of the "Skype as Bandit" Era from Disruptive Telephony
And so it ends... Skype was always always a fun company to write about because they were always a bit of a rogue. The scrappy little startup that took on the megacorps of the telecom industry... and won in so... [Read More]

Tracked on May 10, 2011 at 11:27 AM

» Skype Hits 40 Million Simultaneous Users! from Disruptive Telephony
Congrats to the folks at Skype for hitting over 40 million concurrent users! Today at 2pm US Eastern when I typed "/users" in any Skype chat on my Mac, I got this great message (Windows users should see the count... [Read More]

Tracked on April 10, 2012 at 08:14 PM

Comments

Paul said...
Very useful info and well written, thanks!

ReplyMarch 15, 2011 at 10:49 AM
Comment below or sign in with Typepad Facebook Twitter Google+and more...

(You can use HTML tags like <b> <i> and <ul> to style your text. URLs automatically linked.)

Email address is not displayed with comment.

Name

Email Address

Web Site URL

Post Preview

Disruptive Telephony
Disruptive Conversations
Twitter: @danyork
Voice of VOIPSA
Blue Box