

Disruptive Telephony

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

[Disruptive Telephony](#)

[Disruptive Conversations](#)

[Twitter: @danyork](#)

[Voice of VOIPSA](#)

[Blue Box Podcast](#)

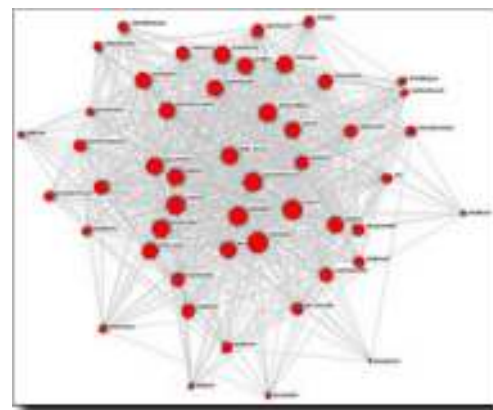
[Voxeo Blogs](#)

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?

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.*

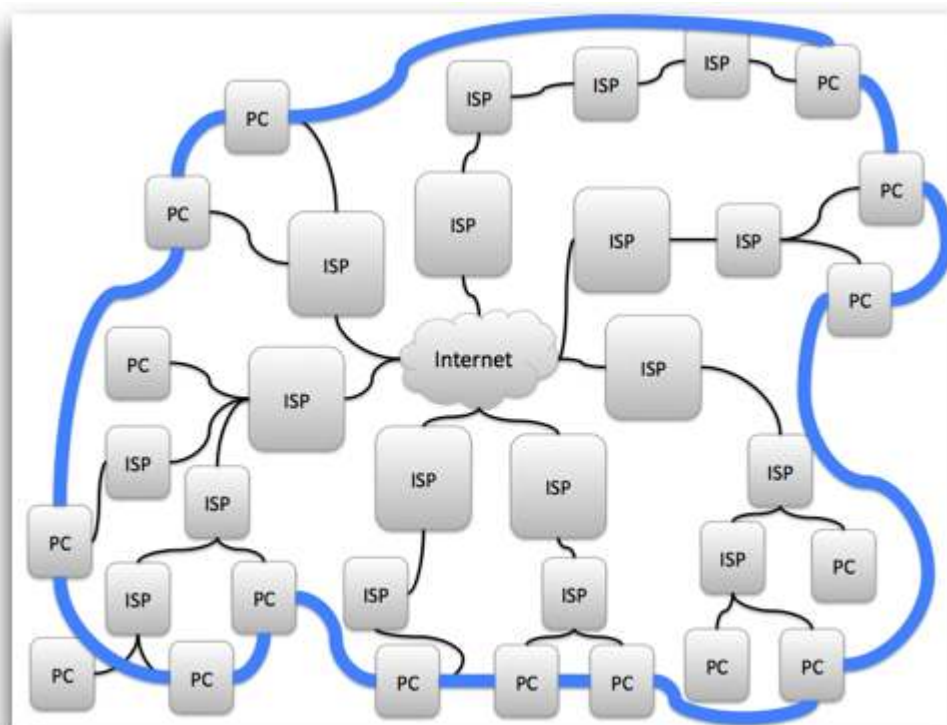
(Well... at least in "pure" P2P networks. We'll talk about hybrids later.)

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](#)*

[Email this](#) • [Add to del.icio.us](#) • [Technorati Links](#) • [Digg This!](#) • [Share on Facebook](#) • [Stumble It!](#)

November 01, 2010 in [P2P](#), [SIP](#), [Skype](#) | [Permalink](#) | [Comments \(0\)](#) | [TrackBack \(0\)](#)

[Favorite](#)[Reblog \(0\)](#) || [Digg This](#) | [Save to del.icio.us](#) | [Tweet This!](#)

Another Hotel Fails To Support Skype - Here's Why Skype's P2P Connection Model Breaks Their System

UPDATE: *When I stayed at this same hotel in August 2010, I no longer had the issue with Skype being blocked. Presumably they got a smarter network monitoring system. While this specific hotel now works with Skype, the same issue will undoubtedly be out there for many other hotels and locations.*

Summary: Hotels restricting the number of simultaneous network connections per user may wind up blocking legitimate usage of Skype. Skype's peer-to-peer network model uses a high number of network connections to synchronize multi-party group chats.

Read on for the full story, network diagrams, etc....

Two weeks ago on a visit to [Voxeo's](#) corporate headquarters in Orlando, FL, I stayed at the [Grand Bohemian Hotel](#), conveniently located only a block or so away. Arriving in the early evening, I checked in, got to my room and immediately plugged my laptop into the Ethernet port to catch up on what had happened while I'd been offline traveling. As is the case in many hotels, I was asked to login and pay through a system from "Nomadix". I did so... and very quickly started to see Skype coming online, my other IM client (Adium) coming online, email starting to flow in and a website coming up.

Then it all stopped.

No Internet connection. Offline. I did all the standard things... disconnect and reconnect the cable. Stop/start the network interface. Nada. Nothing. Dead.

I'd only been on a minute or two but it had seemed to be fine, so I naturally called the front desk who put me through to the tech support team which turned out to be an external company. (Note: Nomadix makes the gateway that is used by this external company to operate the hotel's network. The network is not operated by Nomadix.) They, too, ran through the typical checks, found nothing, and then checked the list of blocked IP addresses - *there I was*.

The technician unblocked my IP address... I saw that I was online again... and then after a minute or so I wasn't. I called back in, spoke with a different technician, had the same experience and stayed on the phone for a good bit investigating the matter.

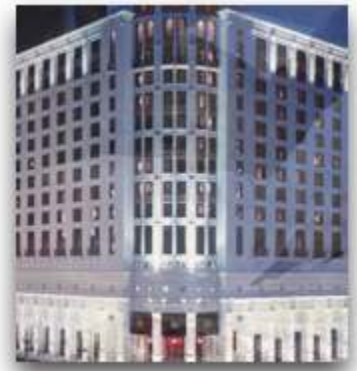
It turns out that:

| *they automatically block IP addresses that generate over 200 simultaneous network connections.*

And here is my dilemma:

| *I am a heavy user of Skype, particularly Skype's persistent group chats.*

Every time I connected, Skype was initiating hundreds of network connections to update all of the group chats that I had open. This, of course, was triggering the rules in the Internet gateway and landing me on the blocked list. If the technician unblocked me (or later testing seemed to



show that every 15 minutes or so the block was released), I would then wind up blocked again after only a minute or so.

The support technicians were all very pleasant and explained that unfortunately the 200-connection-limit was hard-coded into the gateway system and there was nothing they could do (at their level, anyway) to change or set aside that limit.

As a security guy, I do understand some of what the company is trying to do here with the limit. They do have to treat the hotel network as "hostile" to a certain degree. Someone with malicious intent might connect to the network and try to execute a Denial-of-Service attack or send out spam. Or someone might unwittingly be infected with a bot that is commanded to execute some attack. They also want to prevent someone from sucking up all the network bandwidth so that other hotel users receive poor service. Limiting the network connections is one way to potentially try to deal with these type of attacks. Unfortunately, the limits also restrict the legitimate usage of Skype.

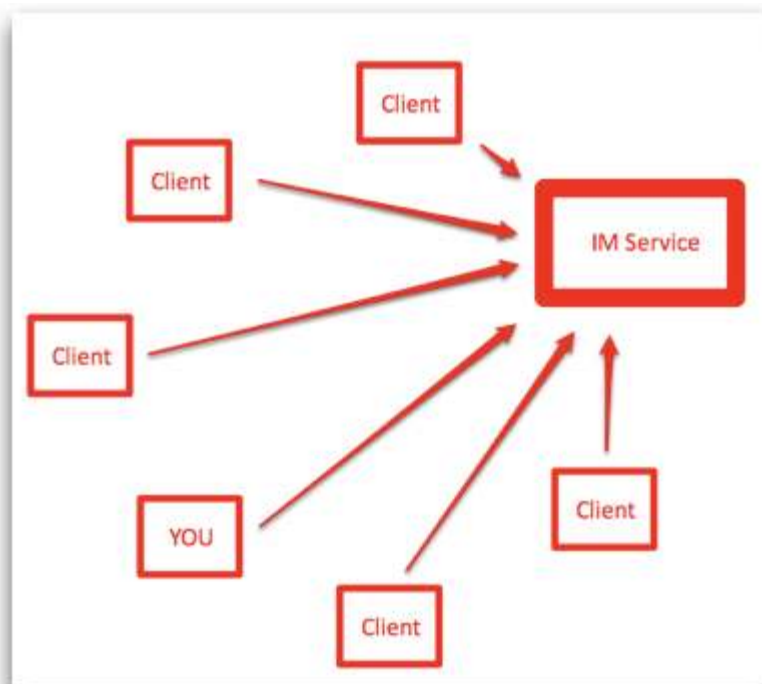
To explain how Skype plays a role here, let's dig a bit more into the way in which Skype's P2P architecture works...

THE BASICS OF PEER-TO-PEER (P2P) ARCHITECTURE

In any given peer-to-peer service (Skype or otherwise), there is a fundamental design difference from "typical" services:

| *there are NO centralized servers.*

If you look at a "standard" instant messaging (IM) service, all of the IM clients connect in to a central server (or group of servers). From a network point of view, it looks like what you see in this image:



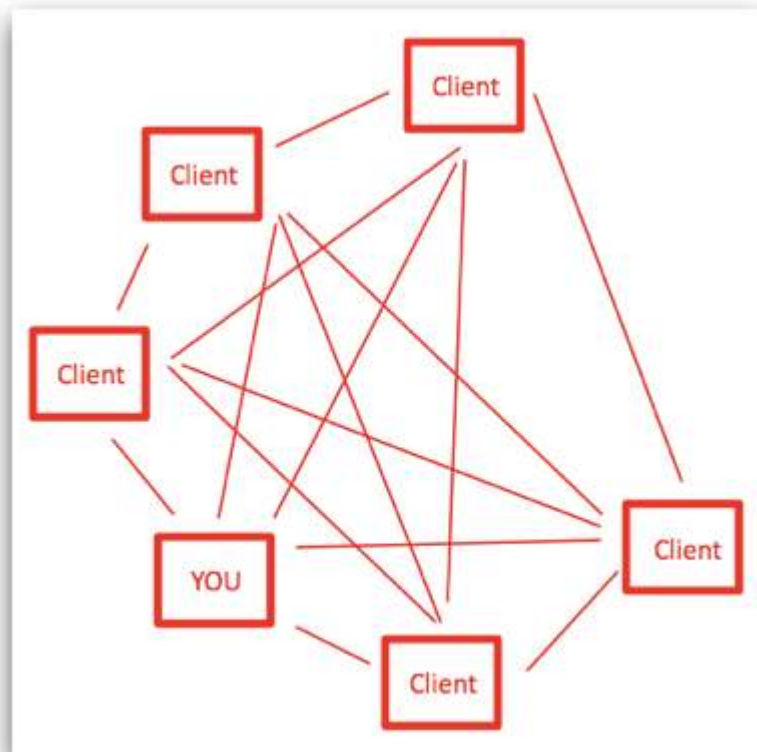
This could be for AOL/AIM, MSN/WLM, Yahoo!Messenger, Jabber, IRC or pretty much any of the other IM services out there outside of Skype.

ALL traffic goes through the central server. If you want to send an IM message to the person sitting next to you, your IM messages are still going through a central server somewhere.

From a network traffic point-of-view, each IM client is opening up a small number of connections to the central IM service. It might only be a single connection, or it might be a couple... but it's only a few. All traffic, both control messages and the messages themselves, travel across those few connections - and all through those central servers.

Over in peer-to-peer land, though, where *there are no central servers* all of the communication occurs through what is

typically referred to as an "overlay network" (or "P2P overlay network" or "P2P overlay"). The overlay network is essentially a "virtual network" between all the nodes in a network. From an architecture perspective, it looks like this:



It's a "mesh" network where all of the service clients are interconnected with all the other clients. (In fact, the term "peer" is typically used instead of "client".) Typically in a P2P network this "overlay network" may be a "distributed hash table" (DHT) using a protocol like Chord or Bittorrent. It sits on top of a systems existing IP connection to the Internet - hence the term "overlay". So this is a P2P network sitting on top of the regular IP network.

SKYPE'S P2P - AND PERSISTENT GROUP CHATS

Caveat: *I am not an employee of Skype and have no real connection with Skype other than having been an user for something like 5 years now. The material below is based on educated guesses - and could be entirely wrong. (And I'd love it if someone from Skype would confirm or deny any of the info here...)*

Skype uses some type of P2P network for communication between Skype clients. When your Skype client comes online, it connects to the Skype P2P overlay network. I don't know personally what protocol Skype uses for its overlay network, but odds are that it is some kind of DHT or similar system. Now, Skype's network is not entirely a P2P network in that there *are* some centralized services that do, for instance, authentication (which were part of the outage back in August 2007), but for the most part it's a big P2P cloud.

Now let's talk about Skype's "persistent group chats". The strength of Skype's multi-person group chats is that you can shut down your computer, travel somewhere, open your computer back up, have Skype reconnect..

 | *and receive ALL communication that occurred in the group chat while you were offline.*

This is a *huge* benefit to an IM-centric organization. If you shut your computer down at the end of the day, or if you are traveling, you simply reconnect and have the complete history of all communication that occurred while you were offline. It works fantastically well for globally distributed teams. I use it heavily within Voxeo and for external teams as well.

The question naturally is... when you are offline, and if there are no central servers...

| *where are the chat messages stored that you get when you come back online?*

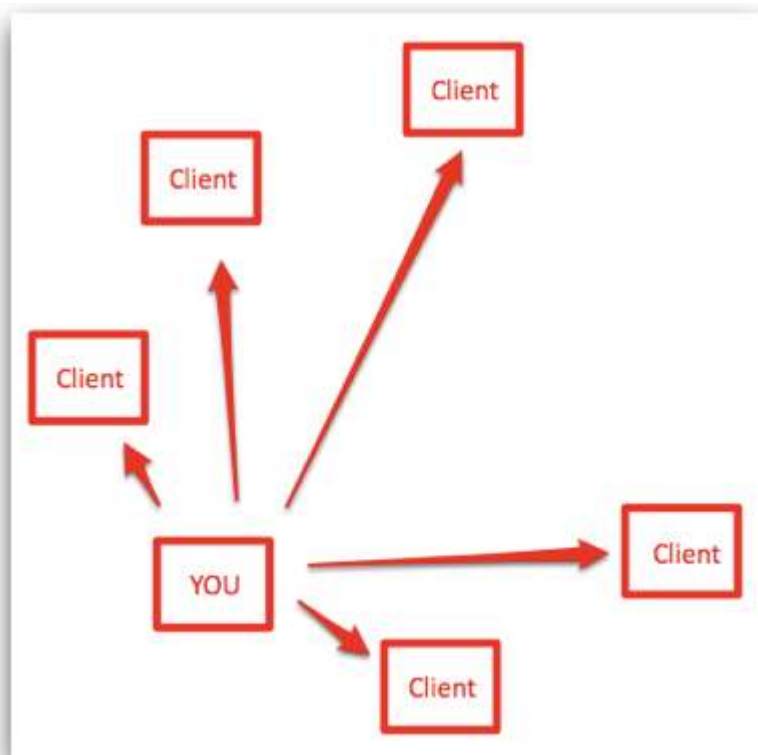
The answer, of course, is:

| *the messages are stored in Skype's P2P overlay network.*

(Do you see yet where this is going and the problem it's going to create?)

So if I am in a Skype group chat with five other people, all the text we type is stored in the Skype overlay network and specifically in a mini-network or "ring" between our 6 nodes. Now... we don't know Skype's P2P protocol to know precisely *how* it is stored across the nodes... but some parts of the text are probably found in all the members of our little ring.

When you have been offline and come back online, your Skype client has to connect out to the others in your mini-network to update your local client with all the messages that occurred when you were offline. Because your Skype client may not know if the 5 other clients were all online during the entire time you were off, your client seems to need to check with *all* of the other nodes in your mini-network. So it looks something like this:



Now, if you look at it from a network traffic point-of-view, *1 Skype groupchat generated 5 network connections*. (And that assumes only a single connection is made per Skype node.) From various discussions and research over the years, the rule seems to be:

| *Each Skype groupchat is going to generate a number of network connections equal to the number of participants in the groupchat, up to a limit of 15 per chat.*

If you have an open chat to one other person, when your Skype client comes online it generates 1 network connection to that person. If you have an open chat to 2 other people, Skype opens 2 network connections. For 5 others in a chat, that's five connections. Ten others in a chat, 10 connections... and so on.

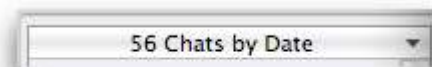
This obviously doesn't scale when you get into very large group chats - I'm in one public chat that has been around for years and has 200+ participants. In those cases, a Skype developer a few years back said in a public chat that in group chats larger than 15 people, your Skype client would connect out to 15 other nodes in the chat to get updates. I don't know if that is still true today, but it seems a logical way to address the scaling issue. Odds are that if you connect out to 15 other nodes, some number of those are going to be online and have enough of the chat history to get your client

up-to-date.

For the purpose of this post, let's assume that is still accurate... and so any chat with more than 15 users generates 15 network connections when your Skype client comes online.

THE CHALLENGE OF THE HEAVY SKYPE CHAT USER

So here's the problem that I believe nailed me at the hotel. If I look at my Skype client right this moment, I have 56 chats open in one window and 20 chats open in another - total of 76. That's actually down a bit because I went through and closed a bunch recently. In scanning through the list, probably 15 of them are 1-to-1 chats that I either keep open because they are people I frequently communicate with or are new chats that I haven't closed - and keeping the chats open lets me very easily see their presence. The rest are multi-person group chats that range from 3-5 people up to 150 or in one case 200 people. Most of them seem to be in the 10-20 person range. Some are long-term chats that I keep open because there is frequent traffic in them, others are short-term chats that have been set up for specific projects or events and will then be closed when the project or event is over.



Without actually going through and calculating the precise number, I'm going to guess that the average number of participants across the 61 group chats is probably around 10-20.

For the sake of keeping the math simple, let's just assume the average number is 10 users. Multiple that by 61 and then add in the 15 one-to-one chats and you have:

| 625 network connections

Oops.

If the hotel blocks a user at *200 connections*, I'm obviously triggering limits. Even if my average is off, or if Skype does something to space out connections over time (which it doesn't seem to do) or to otherwise make connections between users more efficient, I am still probably going to run over that 200 connection limit. My network traffic profile at a high level is going to look like a spammer or DoS attack.

Keep in mind that these are short, quick connections just to sync with the other nodes in the ring created for each chat and get any messages - so we are not necessarily looking at a large amount of *bandwidth*, but we are looking at a large *quantity* of connections.

[NOTE: The next step someone needs to do is to take some *wireshark* captures and generate some pretty graphs of network connections.]

WHAT TO DO?

So now what? What should I as a user do?

1. **DON'T USE SKYPE** - I'm sure someone will suggest this. However, Skype estimates that over 30% of their traffic is business usage. Outside of my own usage, I know many people who use Skype as a significant part of their business communication. Not using Skype obviously is a solution, but not the desired outcome. ***This isn't only about "skype"***. While Skype is the issue in this post, this is a general concern with P2P architectures in general. As an open standards supporter, I'd love to see someone come along with a solution based on P2PSIP that provides similar features - but guess what, it's going to run into similar issues. It's an *architecture* issue - the idea of blocking on some number of connections is based on the old-fashioned client/server model where local clients make only a small number of connections out to dedicated servers. For that model, it may work... but that doesn't reflect evolving usage of P2P networks that are a mesh between nodes.

Today the issue hits Skype... tomorrow it may hit some other cool application that uses P2P for communication.

2. **HAVE FEWER OPEN GROUP CHATS** - It's not clear to me how quickly Skype checks the status of "closed" group chats, i.e. ones that you are still a member of but are not currently displaying. It has to check at some point in

case someone typed messages there, but does it do it on initial connection or launch? (I don't know.)

3. **REDUCE THE NUMBER OF GROUP CHATS** - Obviously this can help address the issue... simply "leave" (versus "close") many of the chats you are in. However, the persistent group chats are one of Skype's great features and enable very powerful collaboration between globally distributed teams. Not really an option.
4. **CHANGE HOTELS** - Of course we as users have the option to find other hotels that don't place the same restrictions... but sometimes we don't have that option.

What can hotels do?

1. **RAISE THE CONNECTION LIMIT** - An obvious solution is raise the number of simultaneous allowed connections... to what number, I don't know... there are trade-offs in trying to block the illegitimate traffic that may be on the network.
2. **PERFORM MORE INTELLIGENT LIMITING** - Applying a hard limit on the raw number of network connections is a rather brute-force approach. Instead the software should look at the quality of the network traffic. Are there are large number of high-bandwidth connections? Perhaps someone is downloading software or movies via some P2P network... in that case maybe they need to be throttled back or limited. Are they smaller connections that may be okay? Can they identify the actual Skype traffic and allow it but block other traffic?
3. **THROTTLE/LIMIT VERSUS BLOCK** - The rules I ran afoul of block your entire Internet access. Too many connections and your link goes dead. Why not truly limit or throttle back the connection instead of terminate it entirely? There's technology out there that can do this type of thing. (Consider, for instance, the idea behind good old [ICMP source quench](#).)
4. **ALLOW TECHNICIAN OVERRIDES** - When I spoke with the technicians and explained what I was doing, the technicians had no options other than to momentarily unblock my IP address. Why not allow them to have a "white list" to which they could add the addresses of certain guests who request special access? It doesn't solve the issue, but it at least would keep certain guests happy.
5. **GET A NEW SOFTWARE SOLUTION** - Obviously to do these steps the hotel may need to look at new software... or a new Internet provider.
6. _____ - *What else do you suggest they do?*

It's 2010 - the reality is that Skype isn't going away... and P2P architectures are continuing to evolve and provide interesting ways to solve communication challenges. The fully-meshed P2P overlay network will continue to be a feature of proprietary networks like Skype as well as standards-based solutions. Travelers want to use communications solutions like Skype... and hotels and their Internet providers need to figure out how to allow the legitimate usage of these tools and services while still keeping their controls in place to block malicious network usage.

What do you suggest? What would you do as a user or as a hotel?

*P.S. This issue has been around for quite some time... [I wrote about another hotel blocking Skype back in 2007](#). Same issue... blocking on **quantity** of connections versus actual network impact.*

If you found this post interesting or useful, please consider either [subscribing to the RSS feed](#) or [following me on Twitter](#) or [identi.ca](#).

[Email this](#) • [Add to del.icio.us](#) • [Technorati Links](#) • [Digg This!](#) • [Share on Facebook](#) • [Stumble It!](#)

April 05, 2010 in [Architecture](#), [Internet](#), [P2P](#), [Skype](#) | [Permalink](#) | [Comments \(7\)](#) | [TrackBack \(0\)](#)

Favorite

[Reblog \(0\)](#) |

| [Digg This](#) | [Save to del.icio.us](#) | [Tweet This!](#)

Of Skype, SIP, P2P and P2PSIP...

Over on Voxeo's Speaking of Standards blog, I put up a post today on:

| [Could Skype realistically replace its P2P algorithm with P2PSIP?](#)



I decided to write it after reading the comments [on Phil Wolff's post last week](#) over on Skype Journal... mostly to talk a bit more about what P2PSIP is and how it compares to what Skype is using now.

While it's interesting to talk about on a technical level - and I admit to a complete fascination with the technology behind P2P networks - the reality is that none of us really know anything about what Skype is up to. :-)

If you found this post interesting or useful, please consider either [subscribing to the RSS feed](#) or [following me on Twitter](#) or [identi.ca](#).

[Email this](#) • [Add to del.icio.us](#) • [Technorati Links](#) • [Digg This!](#) • [Share on Facebook](#) • [Stumble It!](#)

October 19, 2009 in [P2P](#), [SIP](#), [Skype](#) | [Permalink](#) | [Comments \(0\)](#) | [TrackBack \(0\)](#)

Favorite

[Reblog \(0\)](#) |

| [Digg This](#) | [Save to del.icio.us](#) | [Tweet This!](#)

Does the Skype/Mangosoft patent settlement about "dynamic directory service" bode ill for the emerging P2P landscape?

Now that we see some incredibly powerful peer-to-peer (P2P) technology models emerging in the telephony/communication space, will we see that innovation being challenged or delayed by patent lawsuits?



The New Hampshire Business Review reported this week that [Skype has settled a patent lawsuit with Mangosoft](#) for \$2.3 million over a patent apparently related to "dynamic directory service". Now [per the NHBR article](#), it would appear that Mangosoft is fading away as a company and indeed while [the website](#) appears on initial view to be there, [the management team is simply the one CEO](#) and the newest "news" on the web site dates from early 2007. Their [news release](#) about the settlement with eBay is very brief and refers now to "MangoSoft Intellectual Property, Inc." Phil Wolff over at Skype Journal [notes that MangoSoft's SEC filing](#) is also brief (but discloses the amount). Looking back at [MangoSoft's 2007 annual report](#), they are themselves very clear on what they are doing:

BUSINESS STRATEGY

We no longer develop new software products or services. We continue to market, sell and support our software

services. Our strategy also includes seeking strategic business partnerships and distribution channels to leverage our patented technology. All of our business operations are overseen by our sole officer and director, who utilizes third party contractors, as required, to implement the Company's business strategy.

Though I had not heard of Mangosoft until this article (even though I was living in southern NH during their height), I will say that their technology sounds interesting and indeed in reading Mangosoft's patent 6,647,393 on "Dynamic Directory Service" (either at the US Patents and Trademark Office or over on Google Patents) their invention filed back in 1997 does appear to be essentially what we would call today a peer-to-peer distributed directory service, where "directory" is used in the truly generic form as referencing a list of objects of any form (ex. file descriptors, user info, any pieces of information). *[Obvious HUGE caveat - I am NOT a patent lawyer, nor do I play one on TV or the Internet or anywhere else.]* From what I know of Skype's architecture, it would seem that they do use a distributed directory service and so it is perhaps no surprise that they eventually settled.

The question is really - *is this just the beginning of more lawsuits in the P2P space?* MangoSoft's annual report for 2007 shows a debt of \$89 million as of December 2006 and the NHBR articles notes that the trend in operating losses has continued with a \$680,000 loss in 2008 year-to-date. There is obviously an incentive for them to continue on to try to recoup the ~\$90 million that investors have sunk into the company. Beyond this patent, Mangosoft holds several other patents that are related to distributed architectures. It could very well be that this \$2.3 million from Skype will be invested now in future lawsuits against other players in the space. Or perhaps not... perhaps it will simply be distributed to some of the existing investors as the operation fades away. I guess that will largely depend upon how much of a solid case to proceed MangoSoft's investors and sole employee believe they have.

While I am definitely sympathetic to inventors who pursued a new technology but were perhaps too far ahead of their time, I must say that I'm not personally excited to see more lawsuits hitting the industry as we see more and more companies (startups, typically) exploring new ways to build communications technologies based on P2P networks. We're in a fascinating time from a network technology point-of-view, as massively distributed networks are now possible and through systems like Skype and BitTorrent we've seen that they are very possible to create. I'd like to hope that this innovation will continue unimpeded by legal battles... although I realize that that's probably an idealistic dream. Even if MangoSoft does *not* pursue others, over time other larger players will challenge the startups in court should they become more of a competitive threat.

Ah, well, we shall have to see. In the meantime, I guess the good news for Skype is that with their one-time licensing of MangoSoft's patents, they will at least be protected from any further issues in court on these particular patents.

If you enjoyed this post, please consider either subscribing to the RSS feed or following me on Twitter or identi.ca.

Technorati Tags: [skype](#), [voip](#), [mangosoft](#), [p2p](#), [patents](#), [peer-to-peer](#), [telephony](#), [communications](#)

[Email this](#) • [Add to del.icio.us](#) • [Technorati Links](#) • [Digg This!](#) • [Share on Facebook](#)

December 26, 2008 in [P2P](#), [Skype](#), [VoIP](#) | [Permalink](#) | [Comments \(0\)](#) | [TrackBack \(0\)](#)

Favorite

[Reblog \(0\)](#) |

| [Digg This](#) | [Save to del.icio.us](#) | [Tweet This!](#)

Skype and SIP interop - the two sides of the issue raised by Michael Robertson

Should Skype open up it's network to other users? to other networks? Should Skype stop preaching





about "openness" when it's network remains closed?

In the middle of last week, there was quite a little storm raised in the VoIP corner of the blogosphere after [Andy Abramson published a letter from Gizmo Project founder Michael Robertson](#) critical of Skype's openness after [Skype continued to call upon the FCC to open the wireless network to applications](#). (See also [here](#) and [this Skype blog post](#) (and [this one](#)) for background.) Being at ITEXPO last week, I didn't have the chance to blog about this at the length I felt it deserved until today.

First, for some context, here are some of the blog posts last week:

- Andy in VoIPWatch: [Gizmo Project's Michael Robertson Sounds Off](#)
- Phil Wolff asks some pointed questions (of Michael) at Skype Journal: [Michael Robertson says Skype should open to Gizmo Project](#)
- Phil posts: [Michael Robertson Responds](#) where Michael responds to Phil's questions
- Andy responds: [For the Record, Gizmo vs. Skype](#)
- Alec Saunders had a whole cast of characters on [Friday's SquawkBox](#), including Andy, Michael Robertson, Jim Courtney (Skype Journal) and a good number of others.
- Phil posts: [A few thoughts on Skype interop tradeoffs](#)

All of it makes for good reading. It's an important issue.

So I guess I'm somewhat of two minds on this issue.... while I agree with some of Michael Robertson's points, I guess I see the whole issue of "Skype openness" as quite orthogonal to the larger issue of open wireless spectrum. I'll write about both issues at some length below.

This is long. Don't plan on reading it on a crackberry or iPhone. Best get a cup of coffee and read it on a laptop or something like that.

WIRELESS OPENNESS

To put it another way, I completely applaud Skype's letter to the FCC and think that the battle for opening up the wireless networks is probably the preeminent "battle" we who are advocates for an open Internet have before us. The wired carriers are well on their way to being commoditized big, fat, dumb pipes. The telcos started selling "data lines" and then soon the world of IP wound up increasingly removing them from the picture as anything other than a big pipe... and that's not the world they want. They are fighting (and will continue to fight) to retain relevance (and ARPU - Average Revenue Per User) but the IP train left the station a long time ago and the telcos are scrambling to catch up and stay on board rather than just being (some of) the providers of the track.

On the wireless side, not only do the carriers own the track, but they still own the train and they are still driving that train. They control who gets on and off... how fast or slow it goes... what color the train is, etc.

I, for one, don't want that. I want them to give us a solid set of tracks to use... but I want the train to be open to all. I want the wireless carriers to be big, fat, dumb pipes. I want choice. *I* want to be in control.

The carriers naturally don't want to relinquish this control. They see how they missed it on the wired side. They want to keep the wireless walled gardens for as long as they can.

The cracks are appearing... Apple's control over the iPhone and AppStore is a phenomenal crack in the telco walls..... although it ultimately really means exchanging the walled gardens of the telcos for the bright shiny walled garden based in Cupertino, CA. I'm not sure that ultimately is the best for all of us... but it does crack the telco walls. I think Google's Android has more potential... but we'll have to see later this month when those phones first come out.

So with that view, you can expect I applaud Skype's efforts to open up the wireless networks and allow consumers to have a choice of what apps they want to run. I want the *wireless* carriers to be big, fat, dump pipes... give me an IP address on the *mobile* Internet and let me do what I want with it. Sure, the carriers can offer their own services, and maybe if I like them I'll pay for them.... but I want the option to use other products and services - without degradation or

prioritization...

To put it another way, I pay the wireless telcos for *dialtone* now. Once connected, I can call anyone and use any *voice service* over the PSTN. I could use someone else's voicemail if I want (like GrandCentral), although the carrier's offering may be more convenient (and is usually free). But I can call anyone on the PSTN and use any voice service I want. The carriers just provide me dialtone.

I want "IP dialtone". I want a Big, Fat, Dumb Pipe.

So... go, Skype, go!

SKYPE OPENNESS

Yet having said all this, I agree with Michael Robertson that Skype's got its own issues with openness.

I don't like walled gardens. Period. End-of-story.

I don't like telco walled gardens. I don't like Apple's walled garden. I don't like Facebook's walled garden. I didn't like the walled gardens of CompuServe, AOL, Prodigy, Genie, etc. and I rejoiced over time when the open standards of IP tore down those walls and brought about the Internet (with all of its warts) that we have today. [Tangent: I do worry, actually, that we are retreating a bit back into the walled garden world with things like Facebook and Myspace... but [that's another topic I've blogged about.](#)] I can somewhat see some value in walled gardens during the early stages of a product or technology as it reduces the set of parameters/variables and allows the service to be stablized/fixed /improved. But at some point the walls need to be reduced/eliminated. As a security guy, I don't like monocultures... I don't like homogenous systems (where one virus or issue could wipe out the whole system)... I like diversity... heterogenous systems. I don't like single-points-of-failure. I don't like single companies (or small sets) in control. I don't like walled gardens.

I don't like Skype's walled garden.

The PSTN run by the telcos of today does not provide the rich communication experience we want. We need to bypass it and leave it behind and build the massive interconnect of IP communications systems. Players like Skype have a key role in my opinion in building that new infrastructure.

But if we exchange the current PSTN walled garden controlled by the telcos for a new walled garden controlled by eBay/Skype, have we really gained anything?

Sure, it gives us a rich, multi-modal user experience. Sure, it's nice and pretty. Sure, it gives us a central user directory. Sure, it gives us wideband and encrypted audio. Sure, it's cool and all... but it's still a walled garden controlled by a single company.

Ultimately, I would like to see a new voice infrastructure that consists of many different "clouds" all interconnected and able to communicate between the clouds. Skype is one cloud. So are the SIP clouds being run now by various carriers. So are the Voice-over-IM clouds like MSN, AIM, etc. (that try vainly to compete with Skype). So are the various systems being built by vendors all over the place (including all the Microsoft OCS clouds starting to appear within enterprises).

We need to build the interconnect.

Yeah, there are a TON of issues out there that we still need to address to build that interconnect. There's a whole host of security issues... there are billing issues... there are trust issues... there are network plumbing issues. Yes, there are all those issues. But if we are to succeed in ultimately bringing about the rich communication experience we want, we need to make this happen.

And for that, Skype's walls need to come down.... at least a bit.

What we need is that Interconnect from Skype's cloud out to the emerging IP infrastructure. Think about it... Skype right now has a two-way interconnect between Skype's cloud and the cloud we know as the PSTN. It's called "SkypeOut" and "SkypeIn" (or whatever marketing names they are being called now). If you dial my SkypeIn number, you can reach me

on Skype wherever I am. From my Skype client, I can call anyone on the PSTN. The two-way interconnect is already there.

So why not offer the same on the IP side?

Because I work at [Voxeo](#) and we were one of Skype's original Voice Services partners, I already know that Skype has a *massive* SIP infrastructure on the backend to do the the PSTN interconnect. Skype users can even dial a specific +99 number and the call goes from Skype's SIP cloud over to Voxeo's SIP cloud... and it works beautifully.

So one half of the interconnect is already there - although only for limited numbers of partners.

But where Skype already has the infrastructure, why not look at making that capability more accessible? What if someone in Skype could just type "sip:dyork@sipserviceprovider.com" and the call would go out from Skype's cloud to the service providers? This could be a new feature as part of the unlimited calling plans, etc.

How many people would use it right now? Probably only a tiny few... *today*. But suddenly Skype becomes an enabler of the broader post-PSTN infrastructure. New companies can get their services up and running knowing that they can promote them to Skype users and have Skype users get to those services.

Plus, Skype can now connect to all those enterprise VoIP systems being deployed everywhere... so for all those IT managers blocking Skype now but allowing SIP gateways for remote teleworkers using IP phones... Skype can suddenly be that remote softphone being used in the sense that it could connect in to other people on the corporate enterprise system - they just become "sip:" entries in my Skype directory. Skype still is my overall directory and user agent.

And what about the other way? Wouldn't it be great if someone out there on a SIP system could just call something like "sip:danyork@sip.skype.com"? The call goes from their SIP cloud across the Internet to Skype's SIP gateways and into the Skype cloud.

The SIP system user can do this right now... via my SkypeIn number... but they have to use the crappy PSTN. Why not ditch the PSTN and go directly across the IP infrastructure? Hey, maybe some user of a "HD Voice" Polycom phone could call Skype's gateways via SIP and actually wind up talking via *wideband* audio? (Yeah, okay, I'm probably dreaming on that since Polycom supports G.722 for wideband and Skype uses its SVOPC codec.)

I personally would probably wind up using my Skype client **more** for a simple reason that I have a SIP IP phone on my desk... but I'm not at my desk all that much. Wouldn't it be great if I could forward that to the SIP URI of my Skype client? (Which I can do now by forwarding to my SkypeIn # but again I'm going across the crappy PSTN.) Or better yet because I have a SIP URI for Skype it becomes one of the various phones I ring when someone calls that number (it's not, now). The number on my business card would then wind up actually going to my Skype client.

Suddenly Skype can be a player in the enterprise "unified communications" market. People don't need Skype-to-PBX gateways or sacrifice chickens and utter weird incantations to get Skype connections working with open source VoIP systems. Skype users can talk to Microsoft OCS systems... or Cisco IP PBXs... or Avaya's or Nortel's or Mitel's or ShoreTel's or... or... or...

What if... even... I could do a SIP invite to make a *video* call to another system? (Okay, so now maybe I'm *really* dreaming...)

Suddenly people (like Michael Robertson) have fewer reasons to complain. Sure, the Skype client still uses it's own proprietary protocols and codecs for communication *within* the Skype cloud... but you can interconnect.

Suddenly Skype is a leader in building the broader overall next-generation IP communications system. Skype's not a walled garden but rather a player in the larger picture.

Sure, there's a whack of issues involved with doing this. On the technical side, Skype has got to build SIP gateways that could deal with the abuse they would undoubtedly suffer by being exposed on the public internet (like any or all of [the VoIP security tools out there](#)). They have got to make sure such gateways don't become a way to inject spam/SPIT into the Skype network. Skype has got to figure out how to package it... potentially charge for it, etc. And they have to deal

with all the glorious interoperability issues that come with SIP... as the protocol increasingly becomes an unmanageable accretion of all sorts of crap. (And I say that as an *advocate* for the SIP protocol.)

Ultimately, I think that's the kind of openness Skype needs.

Skype needs to provide the same two-way interconnect to the evolving IP communications infrastructure (that is almost all becoming SIP-based, for better or worse) just as Skype provides the two-way interconnect to the PSTN.

Build it and Skype would silence many of the critics of Skype's lack of openness[1] and give Skype a (much-needed, IMHO) hype-boost among the early adopter crowd who also plays with all the other emerging tools. It would be a bold move that would also help Skype gain some credibility and recognition within the larger industry. In my opinion, outside of the technical issues it would go far in so many ways in helping Skype grow - and helping the industry grow.

Will Skype get there? Good question...

[1] *Not all critics would be silenced naturally because someone would still complain that they can't connect their particular client to the Skype P2P overlay network. Or that they can't connect XYZ hard phone to the Skype cloud, etc., etc. But it would silence many of the critics of Skype as a "walled garden".*

Technorati Tags: [skype](#), [sip](#), [net neutrality](#), [open](#), [openness](#), [voip](#), [voice](#), [telecom](#), [telecommunications](#), [gizmo](#), [interop](#), [interconnect](#)

September 22, 2008 in [P2P](#), [SIP](#), [Skype](#), [Telecom Industry](#), [VoIP](#), [Wireless](#) | [Permalink](#) | [Comments \(1\)](#) | [TrackBack \(2\)](#)

Favorite

[Reblog \(0\)](#) |

| [Digg This](#) | [Save to del.icio.us](#) | [Tweet This!](#)

Want to understand Peer-to-Peer SIP (P2PSIP)? Listen to this podcast...

What if we could design SIP-based VOIP systems... *but without any servers*? What if we could have SIP endpoints just communicate with each other and "self-organize" into networks? What if we could essentially build an open standards-based version of Skype? How would it work? Who would use it? How would we secure it?

Those are all questions we discussed in [the Squawk Box podcast / interview I did with David Bryan on July 10th](#). David is the co-chair of the IETF's P2PSIP Working

Group and also the CEO of SIPeerior Technologies. It was a great interview where we covered all these questions and much, much more.

P2PSIP, to me, represents one of the most exciting new directions for SIP research and is something I'm definitely following closely. [I wrote about my interest in P2PSIP clouds \(and connecting them to larger clouds\) at some length over on Voxeo's *Speaking of Standards* blog...](#) it's all about *clouds* of SIP communication... and how we weave them all together. It's a fascinating time.

If you'd like to understand what P2PSIP is all about, please do definitely check out [the Squawk Box podcast...](#) and then, if you are so inclined, head over to [P2PSIP.org](#) to find links to learn more and download code...

Technorati Tags: [sip](#), [p2p](#), [p2psip](#), [telephony](#), [peer-to-peer](#), [voip](#), [ietf](#), [dan york](#), [david bryan](#), [squawkbox](#), [sipeerior](#), [cloud computing](#), [clouds](#)

July 16, 2008 in [IETF](#), [P2P](#), [SIP](#), [VoIP](#) | [Permalink](#) | [Comments \(0\)](#) | [TrackBack \(0\)](#)



[Favorite](#)[Reblog \(0\)](#) || [Digg This](#) | [Save to del.icio.us](#) | [Tweet This!](#)

Additional thoughts on Skype and hotel networks - there's issues on both sides...

To my immense surprise, [my article yesterday](#) about my challenges with Skype and my hotel Internet connection just [hit TechMeme today](#), so welcome, anyone who is coming my way from there. But that also prompted me to want to offer up some additional thoughts on the subject.



First, I'm actually quite annoyed at the Best Western here in Ontario, CA, for essentially blocking Skype by virtue of their network security traffic policies. If travel shall bring me to Ontario, CA, again, you can be pretty sure that *I will not be staying here*. Skype has become an important communication tool for me and <cue violins>was the way I was intending to call home and stay in touch with my family</violins>. Skype has worked great at the hotel I was at earlier in the week in Phoenix and in fact at every other hotel I've been at lately. I do intend to contact Best Western to express my dissatisfaction at being unable to use the program.

Having said that, as a security professional *I do understand WHY* the security team at the Internet provider to this Best Western hotel has the policies in place that they do. As [Phil Wolff commented](#), Skype's launch "can look like the beginning of port scanning or a bot-gone-wild". Given that this provider is dealing with hotel rooms where random strangers are connecting who-knows-what onto the network, they have to be extremely vigilant (especially because customers like me while complain quickly if Internet access is slow/unavailable). The more I think about it, hotel networks are really an absolute *nightmare* from a security point-of-view. You have no way to enforce virus protection, people can put all sorts of machines in all sorts of states onto the network, systems with spyware can easily be scanning/attacking your network -it's really pretty crazy and I'm glad that I'm not involved with running such a network! (Although the security geek in me would admit that the aggregate data they must get from network traffic would probably be fascinating.) However, there is probably a compromise out there where the ISP can tune its filtering rules so that if it sees such traffic and can identify it as Skype traffic, it can *not* trigger the MAC lock-out.

Which brings me to the final point that there's a lesson here for anyone developing P2P apps, or I suppose any other apps that have a similar traffic profile. If the apps generates traffic that looks like a bot or port scan, odds are that it will be blocked in some places like this one (and the hotel Phil was at). It would be great if developers could take that into account and either: a) naturally put in some kind of rate throttling; or b) perhaps provide a "hotel mode" where it throttles back the number of sessions to some (perhaps user-settable but with a default) value. This of course would make it longer for things like presence information to appear, but would at least let you continue to operate the program without triggering the network security alarms. Of course, you'd have to change to that mode, which many people would forget to do and wind up being locked out, but it might be an interesting "advanced" option for those who know what to do with it.

Any other "lessons learned" you see here?

Technorati Tags: [P2P](#), [security](#), [skype](#)September 30, 2007 in [P2P](#), [Security](#), [Skype](#) | [Permalink](#) | [Comments \(4\)](#) | [TrackBack \(0\)](#)

[Favorite](#)[Reblog \(0\)](#) || [Digg This](#) | [Save to del.icio.us](#) | [Tweet This!](#)

How using Skype disrupted my hotel Internet connection and locked me out

UPDATE: I have now [posted some additional thoughts](#) about this issue.

It's been a frustrating time here at the hotel in Ontario, CA, where all I've been trying to do is use the Internet connection. I'm staying at the Best Western and did so largely because they advertised free high-speed Internet (they were also cheaper than others). First annoyance was discovering that I was too far away from their APs to use wireless, but since I had an ethernet cable I just plugged into the wall jack and expected to get access. The very first time I connected, I *did* get an IP address and could see an entry in my routing table for the default gateway. However, I couldn't ping it.



Being rather used to network troubleshooting, I did the usual things... bringing the interface up and down, disconnecting and re-connecting the cable. I even went to the hotel lobby and got a new cable in case the issue was with my portable/retractable cable.

Nothing. No net.

In desperation I did the thing that tech support always tells you to do but I avoid... reboot. Nothing.

So finally this morning I got on the phone to the Best Western tech support and after waiting, oh, 20 minutes or so I got through to a tech and ultimately we figured out the problem:

Skype!

More specifically, all the bizillion connections that Skype was making out into the P2P cloud. The tech reset the switch and asked me to connect again and his immediate response was "Whoa! Something on your computer is generating an incredible number of sessions out to the Internet! You are tripping our filters and it is blocking out your MAC address." With him on the phone, we tried some experimentation. I shut down Skype, at which point he said I was generating much more normal traffic. As soon as I launched it again, he noticed a very large jump in the number of session connections I was establishing. He said it was something like 396 sessions he was seeing coming from my computer. He also said that I'll keep being locked out of their system if I keep Skype running.

So I shut down Skype. Which, of course, is annoying. Part of why I wanted to use the high-speed Internet is to use Skype for IM and for voice/video calls.

I find it a bit odd that Skype was generating *so much* extra traffic, but then again I *am* pretty much always connected into several persistent group chats and had maybe 8 or 10 individual chat windows still open that I'd left open from when I'd last been chatting with the person. (The Mac Skype client makes this easy to do, but I'll write about that sometime.) The persistent group chats, especially, do generate a good number of connections as they link out into the P2P cloud. Perhaps if I closed all of those windows and killed off all my individual chat windows Skype might have behaved better. (Or perhaps not, I might have had to *leave* the persistent chats in order for Skype to stop making those connections.) I don't want to try it out, because I do want to keep my Internet connection up right now.

In any event, should you be at a hotel and find yourself unable to connect... it might be a P2P app like Skype tripping off the hotel's filters and blocking your access. Fun, fun, fun....

September 29, 2007 in [P2P](#), [Security](#), [Skype](#) | [Permalink](#) | [Comments \(18\)](#) | [TrackBack \(4\)](#)

[Favorite](#)

[Reblog \(0\)](#) |

| [Digg This](#) | [Save to del.icio.us](#) | [Tweet This!](#)

-
- [Disruptive Telephony](#)
 - [Disruptive Conversations](#)
 - [Twitter: @danyork](#)
 - [Voice of VOIPSA](#)
 - [Blue Box](#)
 - [Voxeo Blogs](#)
-