VoIP and POTS Integration with Asterisk

by John Todd
01/22/2004

VoIP and POTS Integration with Asterisk

In my last article on Asterisk, I demonstrated how to build a very basic two-line PBX (Private Branch eXchange) with Asterisk and two SIP clients. Asterisk is an open source PBX replacement system, which does in software what many expensive PBX systems do in custom hardware. Voicemail, voicemail/email forwarding, call forwarding, voice menus, multi-ring -- these are just a few of the hundreds of features that Asterisk offers.

In the previous example, I created a small system that allows two people connected to the same Asterisk system to talk to each other, but obviously this is insufficient for most business needs. The goal of a PBX is not just to connect people inside of the office to each other, but also to connect those people to external endpoints, traditionally on the PSTN (Public Switched Telephone Network -- the phone system.) As Asterisk is a VoIP platform (or an Internet PBX: IPBX) it would also make sense if our PBX was able to accept calls from SIP clients or gateways elsewhere on the Internet, as those are growing methods of call delivery. Both inbound and outbound calls from the PSTN and from the Internet (via SIP) must find their way to the right destinations.

This article will demonstrate how to configure our system so that it can receive calls from other SIP clients elsewhere on the Internet. I'll also cover the installation of an FXO card into our Asterisk server, to enable call termination and origination to the PSTN on our existing analog line. Finally, I'll configure our system to use an Internet-based long-distance service provider at a fraction of the cost of traditional long distance.

Handling SIP calls from Other IPBXes

Recall that we have our Asterisk server running, using an SIP phone on the desk or as a software client on a laptop. So how do we receive inbound calls so that people can start calling you directly over the Internet, if they too have a SIP-capable phone system? One of the nice things about SIP is that it was designed to understand alphanumeric usernames, not just phone numbers. If we specify a SIP address such as "buckaroo@pbx.banzai.com," then our SIP phone will (in the simplest case) try to open a connection to the machine represented by the name pbx.banzai.com, and then attempt to call the extension "buckaroo". It gets more complex than this with some designs, but we'll leave this at the most basic configuration for the purposes of our discussion.

In order to implement this type of inbound dialing translation on an Asterisk server, we need to take all of the inbound calls in `sip.conf` that are not specifically associated with a SIP peer and send them to a context in our dialplan. The `[general]` stanza in `sip.conf` contains a default context to which all "unknown" SIP calls are delivered. Once inside of that context, we can convert names to numbers and redirect the call back into the normal dialplan so that the appropriate device will ring. This is very similar to the way email addresses are sometimes aliased on systems. Helpfully, the syntax for dialing an SIP number looks identical to that of an email address ("user@domain.name.suffix").

Handing Off Calls to the Analog Phone Network via SIP

The PSTN is the beast that probably delivers most of your calls right now. Moving to VoIP is all well and good, but for the bulk of our voice traffic, we still are going to want to hand off the call to the "normal" phone system. To do this for non-local calls, we'll use the services of an IP Communications Service Provider (IPCSP). For local calls that don't cost us anything, or for geographically-specific calls (911, 411, 0, etc.), we'll hand these to an analog connection on the existing telephone line. Also, if our IPCSP is unavailable for some reason, we'll fail over to the analog line.

There are a variety of IPCSPs that are starting to emerge as delivery mechanisms to translate our VoIP call into something that is connected to the PSTN. Once one is picked, the dialplan is modified to send connection requests (usually long-distance or international calls) to the SIP server of the IPCSP. Since we know the IP address and some other data beforehand about the remote gateways, Asterisk is fully capable of sending calls out to those providers via SIP.

Picking an IPCSP

Choosing an IPCSP (formerly known as ITSPs, Internet Telephony Service Providers, but that acronym is being phased out) is a hit-or-miss process, much like choosing an ISP was back in the dark ages of the commercial Internet. Some providers have developed fairly decent offerings, but each may have restrictions on the product that they actually sell. You must read the very fine print to determine if it will work with your setup, both contractually and technically. Some providers put very strict usage limits on their accounts that forbid "business" use of their systems or revoke many of the privacy rights you would expect from a normal telephone service. VoIP is **not** your normal phone service, and you should not treat it as such -- read those contracts carefully. However, many people find that they are willing to sacrifice some consumer rights in order to get extremely inexpensive phone calls.

As an example on rates: Iconnecthere.com's "North America 1000" plan works out to about 1.1 cents a minute. *Note: I do not endorse Iconnecthere.com, other than saying that their service has worked for me for some time, and I wanted one real example for this configuration.* Iconnecthere.com allows individuals to purchase long-distance service through SIP connections on a single account. There are other providers of long-distance termination service via both SIP and IAX (an Asterisk-specific protocol not covered in this example).

Beware that many providers do not support, or will actively block, connections by SIP devices other than those that they sell. Vonage, one of the most popular VoIP service providers, does not even give users their passwords or connection details, and locks the equipment that they sell as an SIP adapter. They even go to the length of locking the equipment (Cisco ATA-186) so that it cannot ever be reused for other purposes. Vonage certainly does not work with Asterisk in any native SIP modes. Be extremely specific in your discussions with any SIP long-distance provider and ensure that they support Asterisk natively.

Some providers (including Iconnecthere.com) have add-on services, such as assigning you a phone number in a specific area code that they route to you for inbound calls. I do not cover that type of example here, but you can find examples of configurations for that in my online example files.

Analog Equipment

Most small businesses or homes have what are commonly known as POTS (Plain Old Telephone Service) lines, which are usually wired to an RJ-11 wall receptacle. Our example covers connecting one of these lines into our Asterisk system, enabling us to make calls outbound on your POTS line (in certain circumstances), to receive calls from that circuit, to reroute them to one of the applications that is part of Asterisk, or to route the call to a SIP phone for you to answer.

In order to make Asterisk talk with the analog world, an adapter of some type is required to interface the POTS line to your Asterisk IPBX system. The type of line that we have coming in from your wall jack is normally called an FXO line, which stands for "Foreign Exchange Office" line. There are a variety of vendors that are now selling FXO gateway products that are standalone units. These devices have two receptacles: POTS line and Ethernet. The device then translates analog calls into packets, and vice versa. These devices tend to be fairly expensive, starting in the $300 range for standalone units. These devices create SIP calls that your Asterisk server would need to handle much like any other inbound SIP call and redirect appropriately to end stations.

Getting Calls from Your Existing POTS Line

A more cost-effective solution for single-line installations is the X100P from Digium, which is essentially a PCI-based modem card with carefully engineered firmware that allows the device to handle voice data. The cards are currently in the $100 range and work very well with Asterisk, and since Digium's programmers are focused about 50% on the Asterisk project, each card purchased helps support Asterisk. The X100P is the card that we will use in the configurations below, as this card has excellent support in Asterisk (since the Digium staff serve as the maintainers of the majority of the Asterisk code base). Asterisk also supports some non-Digium modems that can use voice, but trust me -- don't waste your time. Just buy the X100P, and you'll end up with two or three more useful days in your life that you can use for something other than relearning `AT` commands.

In the example below, inbound calls from our analog port are routed to an extremely simple "ring-all" command. Most businesses choose to put a more complex front-end interactive voice response (IVR) system in place, so that the caller hears different recordings at different times of the day or is passed to alternate menus based on touch-tone sequences. As

I have said before, the trick with Asterisk implementations is knowing what not to include, so for this version I will abstain from demonstrating any of those features and hold those techniques for a future article. Inbound calls on the analog line will simply ring the two previously-defined SIP phones and then go to voicemail if there is no answer.

Making Calls Through Your Existing POTS Line

The electrical opposite of an FXO card or device is an FXS (Foreign Exchange Station) device, which accepts dialtone. A normal phone, as an example, is an FXS device, and it is controlled by the FXO line. If you want to truly make your conversion to VoIP as painless and transparent as possible, you could get an Ethernet-to-FXS converter system, into which you would plug your existing phone devices so that they would ring when called. The most commonly found type of device doing Ethernet-to-FXS conversion today is the Cisco ATA-186 unit. However, the market is starting to mature, with several competitors at a much lower price point, so it is perhaps worth your time to investigate the devices from other vendors (see list at the end of this article).

There are also PCI-based cards (from Digium and others) that will allow the direct connection of phone stations into RJ-11 plugs on the back of your server, which may offer some additional features for the real hardware hacker. A truly "transparent" conversion could be made to VoIP with an FXO card and an Ethernet-to-FXS converter of some type -- the users of the system may not even know that their conversations were going out a VoIP channel, since all of the equipment with which they are familiar would remain on the desktop and their phones would ring just as they always have on inbound calls. Once you have familiarized yourself with Asterisk, you may want to carefully gauge the implications of a "guerilla VoIP" install, in which you change out your home phone without the other users knowing of the conversion. Be careful with this, though; many spouses/housemates are not forgiving of configuration errors.

It may be the case that you do not want to retrofit your existing analog equipment, and you may want to go directly to an SIP "hardphone" like the Cisco 7960, Grandstream 102, or SNOM 200. These devices look and act like phones, except they have an Ethernet jack on the back and use TCP/IP over Ethernet as their transport protocol. These are a complete solution for the desktop, but tend to be slightly more expensive per line than the Ethernet-to-FXS conversion devices. The hardphones also may not allow you to use your existing telephone hardware, such as wireless phones, headsets, or speaker phones.

Serving the purpose equally well would be an SIP softphone, several of which are available at no cost (such as from Xten) and provide the same speak/listen features as a phone, except that one's laptop or desktop workstation becomes the "phone" part.

For the sake of brevity, we will not cover the configuration of those devices, cards, or software in this article, as hopefully you were able to get an SIP client (hardphone or softphone) working from the previous article on Asterisk. The configuration specifics in Asterisk are basically identical no matter what device you have acting as a User Agent (UA) that sits in front of the user: analog converter, hardphone, or softphone. All of these devices and methods will allow the user to dial, speak into a microphone, and hear sound transmitted from a distant station -- it is just the price and specific hardware technology that differs in their implementations.

Configuration

We're going to put some very bare-bones examples of configurations in place to support an X100P FXO card, two SIP clients, and a single IPCSP account for outbound VoIP-based long-distance termination. The configurations below are fully functional and duplicate many of the settings and methods used in my first Asterisk article, where appropriate. To conserve space, I have deleted or abbreviated many of the commentary sections which were outlined in my first article -- see those examples for more full descriptions of call flow.

My apologies to non-North-American readers; I chose NANP (North American Numbering Plan) dial syntax due to my familiarity with that plan, and also due to the fixed-length-digit lengths of dial strings. Some nations have variable length dialplans, which is certainly possible within Asterisk, but requires significant work and may be more confusing.

Don't get scared with the size of `extensions.conf` -- there are a lot of comments in there. My suggestion would be to run a `grep -v ";" extensions.conf` to get a real idea of how large the file is; it's quite small once those comments are removed. A corollary to this is that one should never embed semi-colons at the end of configurations lines in `extensions.conf`, even though it is possible to put comments there. If you ran a `grep` as I describe above, the search would give you confusing results, because lines with valid settings in them would be excluded.

Installing the X100P Card

Once your X100P is installed in a proper slot, you will need to configure the two kernel drivers, which allow this new Zap card to communicate with your Asterisk system. When you boot your system, make sure the card is recognized by your machine. To verify this, look in the boot output (`/var/log/messages`) for instances of the word "Wildcard," like:

```
Oct 17 03:42:43 pbx1 kernel: Found a Wildcard FXO: Wildcard X101P
```

Now we need to download the Zaptel software package from the Asterisk CVS, which contains all of the drivers for the Zap family of cards. Hint: if you are experiencing problems with echo on your analog calls, you may wish to uncomment the `KFLAGS+=-DAGGRESSIVE_SUPPRESSOR` line and run `make clean; make; make install` -- this enforces a more rapid echo interceptor for analog circuits.

To download, compile, and install Zaptel:

```
# cd /usr/src
# export CVSROOT=:pserver:anoncvs@cvs.digium.com:/usr/cvsroot
# cvs login [the password is anoncvs]
# cvs checkout zaptel
# cd /usr/src/zaptel
# make
# make install
```

You may need to re-compile Asterisk to incorporate the new libraries.

```
# cd /usr/src/asterisk
# make clean
# make
# make install
```

Now, manually load the drivers for the X100P Zap card:

```
# modprobe wcfxo
# modprobe zaptel
# ztcfg -v
```

You should see a "1 channels configured" message if this worked as expected.

Now we need to ensure that the card is not sharing any IRQs with other devices. This is critically important, since these cards generate huge amounts of interrupts during use, and any conflicts with other devices will result in jittery voice and overall poor performance. Look in `/proc/interrupts` to ensure that `wcfxo` has an IRQ all to itself. If it is sharing an IRQ, move the card to a different PCI slot and see if that resolves the conflict.

Once the card is successfully installed and the drivers loaded, you will need to configure your `/etc/modules.conf` file to automatically load the two modules and run `ztcfg`. Typically, the "make install" in the zaptel directory should modify this file and also modify `/etc/modules.conf` appropriately, but I have included minimal configurations below. If your files seem to contain more lines than this, don't worry -- the Zaptel installation scripts put lots of extra lines in there to deal with hardware that you may not have. You can ignore those lines that don't seem to apply.

Configuration Files

View the configuration files.

Note that these examples do not use abstracted dialing statements, and tend towards some redundant (but descriptive) dialing paths. This is intended to be a functional description for the user who wishes to see a clear view of the way Asterisk works. As one becomes more advanced in the development of dialplans, it will be obvious that a generic context for dialing outbound calls should be created, but this introduces significant complexity of description, so I will avoid that method for now. Note that the "comment" characters are different in the first three files than in the other files ("`#`" versus "`;`").

References

- Asterisk
- Asterisk Manual (pdf)
- More configs

- More links

Hardware

- Digium X100P
- Cisco ATA-186
- Sipura SPA-2000
- Grandstream 102
- SNOM 200

Software

- Xten Xlite

Special thanks to Tilghman Lesher (for editing and review) and Mark Spencer (for quick fixes to bugs).

John Todd is a networking and VoIP consultant, specializing in Asterisk implementations.