The Asterisk PBX as a Linked Repeater Controller

By Steve Rodgers, WA6ZFT

Revison 0.3 10/21/2005

This article describes the Asterisk app_rpt application, and how can be used to link repeaters together over the Internet. Asterisk combined with the app_rpt application makes a very flexible and powerful repeater controller capable of controlling 1 to a dozen or more separate repeaters and remote bases separately and simultaneously at the same physical site. We support full duplex linking over VOIP from one Asterisk site to another, autonomous remote bases (more on this later), and a VOIP autopatch. Our solution employs telephony hardware (not sound cards) to achieve this multi-port operation in a flexible and efficient manner.

Each repeater or remote base is treated as a "node". What this means is that it is capable of being operated by itself as a standalone repeater, or can be connected to from other nodes in the system. Each node can handle multiple connections from other local or remote nodes and has conferencing capability built-in. The number of simultaneous connections is only limited by the available bandwidth, and CPU horsepower.

Each node is assigned a unique node address. These node addresses can be any DTMF digit sequence (0-9) from 1 to 8 digits. Node addresses lengths can also vary from site to site; in other words, you can have a system with 2 digit, 4 digit and 8 digit addresses as long as their beginning digits are not the same. Each PC supporting one or more nodes contains a local list of nodes which can be reached from it. The list is a table translates node addresses to IP addresses of the remote node. The node to IP address list is configurable by the administrator of the nodes present at a given physical site. To secure the VOIP links, you may employ source IP address checking and/or MD5 or public/private key authentication. For most applications, source IP checking will provide adequate security while minimizing complexity.

The repeater controller function is performed entirely in software. All audio going to and from various endpoints remains in digital form until it is absolutely necessary to convert it to an analog signal. The repeater control program "app_rpt" is known as an "Asterisk application". It is written in C as is most everything else in Asterisk. app_rpt and Asterisk multi-threaded. Multi-threading is what allows us to run several repeaters and or telephone connections simultaneously on1 PC.

All software and hardware designed in the project is licensed under the GNU Public License (GPL). Basically, this means the software and hardware designs are open source and may be used by anyone provided the terms of the GPL are followed. For details on the GPL, visit www.fsf.org.

Brief History Lesson

Much of what makes this project possible originated from the Zapata Telephony project at www.zapatatelephony.org. and the Asterisk PBX project at www.asterisk.org.

The “Zaptel” project provides an inexpensive means to connect PC's to legacy TDM (time domain multiplexed) hardware through the use of quad T1 PCI interface cards. The design for these cards are “open hardware” can be downloaded from the site. Zaptel's creator, Jim Dixon, was tired of paying too much money for commercial off-the-shelf T1 adapter cards, and that price pressure is what spawned the Zaptel project.

The Zaptel hardware also originally came with open source drivers for the BSD operating system. These drivers enabled telephony applications to be written for the BSD operating system. The Zaptel drivers were not just “bare metal” drivers. They had DSP code in them to do tone generation, and DTMF decoding. One thing to note is that the Zaptel does not manufacture any hardware, but there are firms which manufacture the Zaptel boards such as Govarion (www.govarion.com).

The Asterisk PBX project (www.asterisk.org) is an open source soft PBX which supports traditional TDM telephony hardware as well as the new VOIP standards. Early versions of Asterisk did not support traditional TDM hardware or have tone generation and DTMF decoding support. During the early stages of Asterisk's development, it only supported a frame relay interface. When Mark Spencer found out about the Zaptel project, he took the source code for the Zaptel drivers, and merged it with Asterisk's code base. This greatly improved the usefulness of Asterisk because now it could be used with all of the TDM hardware that was commonly available. Contained within Asterisk are the Zaptel DSP algorithms for Tone generation, DTMF decoding as well as DSP routines for Echo Cancellation, Audio coding and decoding (CODECS). and Audio Mixing.

Mark Spencer's firm, Digium (www.digium.com) , also went a step further and designed additional boards using the Zaptel concept which did not require the use of a channel bank and which are more cost-effective for small installations. The result of this effort was a PCI 4 port line card model TDM400P , and single port T1 interface card model TDM100P.

Asterisk as a Repeater Controller Platform

Asterisk and Zaptel combined with support for the most commonly used VOIP protocols is what enables us to put together a repeater controller application unlike any other currently available. The Asterisk software is what is known as a “Hybrid Soft Switch” What this means is that Both legacy TDM technology and VOIP technology can can be intermixed and used together. Asterisk runs under Linux (www.linux.org). Extensive documentation on Asterisk can be found at www.voip-info.org.

Asterisk uses the concept of “channels” to describe connection endpoints (e.g. a telephone set or a telephone line). Channels can be “bridged” to establish a connection from one channel to another. Channels are described by a technology name. For example TDM channels are named “Zap”, and IAX2 (Inter-Asterisk eXchange, version 2-- pronounced eeks) channels are named "Iax". For the repeater linking application you only need to use two channel types: Iax and Zap. Asterisk handles call routing, and codec translation (transcoding) between channel types.

It is interesting to note that running the app_rpt repeater application does not in any way inhibit the usefulness of Asterisk as a fully functional PBX. For example, the author runs one repeater node and a 20 extension PBX simultaneously on the same Asterisk Box.

Interfacing

There are 2 ways to interface a radio system to Asterisk. The first uses FXS ports on an existing Asterisk PBX with a simple one channel analog radio interface board (ARIB), and the second uses a custom PCI card we designed which takes the radio interface signals from up to 4 separate radios, digitizes them, and sends them to the Asterisk drivers.

If you have and existing PBX with spare FXS ports, and you are onlu interfacing one repeater or remote base, the ARIB soluton is the best way to proceed. If you are putting up a new Asterisk system specifically for the purpose of interfacing to one or more repeaters, than the Quad Radio PCI card is the best solution.

This article will focus on using the ARIB board as prior revisions had no knowledge the PCI solution, and there are a separate set of documents covering the Quad Radio PCI board on our website.

Zap channels used in the repeater linking application are assigned to FXS (foreign exchange station) hardware interfaces. FXS interfaces look like a CO (central office) line card, and provide battery to power a telephone set. There are currently two accepted ways of adding FXS channels to an Asterisk PC platform.

If you only need to interface 1-3 repeaters, then the Digium (see sources at the end of this article) TDMx0 PCI interface card will provide from 1-4 FXS ports depending on how many FXS daughterboards you purchase. You will need to purchase at least two ports for your TDMx0 every repeater you wish to control at a site, and one port for every remote base at that same site. The 'x' in the model number indicates how many daughterboards are present (and what type they are) on the main PCI card. For example, a TDM20 has 2 FXS daughterboards.

If you need to interface to more than 3 repeaters at the same site , then it makes more economical sense to purchase Digium's TE110P single port PCI T1 adapter and a 24 port FXS channel bank. With regard to channel banks, the CAC FXS Access Bank I has been extensively tested by the developers of the project, and is one of the most cost-effective solutions on the used market. See the sources section at the end of this article for a suggestion on where to procure CAC channel banks.

As mentioned above, Zap channels interface to TDM hardware. The physical connection From Asterisk to your repeater is done using 2 zap FXS channels for each repeater you need to interface. One of the Zap channels is for transmit and will be called the "TX pair", and the other will be for receive and will be called the "RX pair". This is known in telecom industry parlance as a "4-wire" interface.

The TX and RX pairs are connected from the FXS hardware interface to the ARIB (analog radio interface board). The ARIB is available in board blank form from the source at the end of this article. The ARIB converts the 4-wire connection to COR, PTT, PL, TX-audio and RX-audio to facilitate connection to your repeater. An example of a 1 repeater system is shown in figure 1 below:

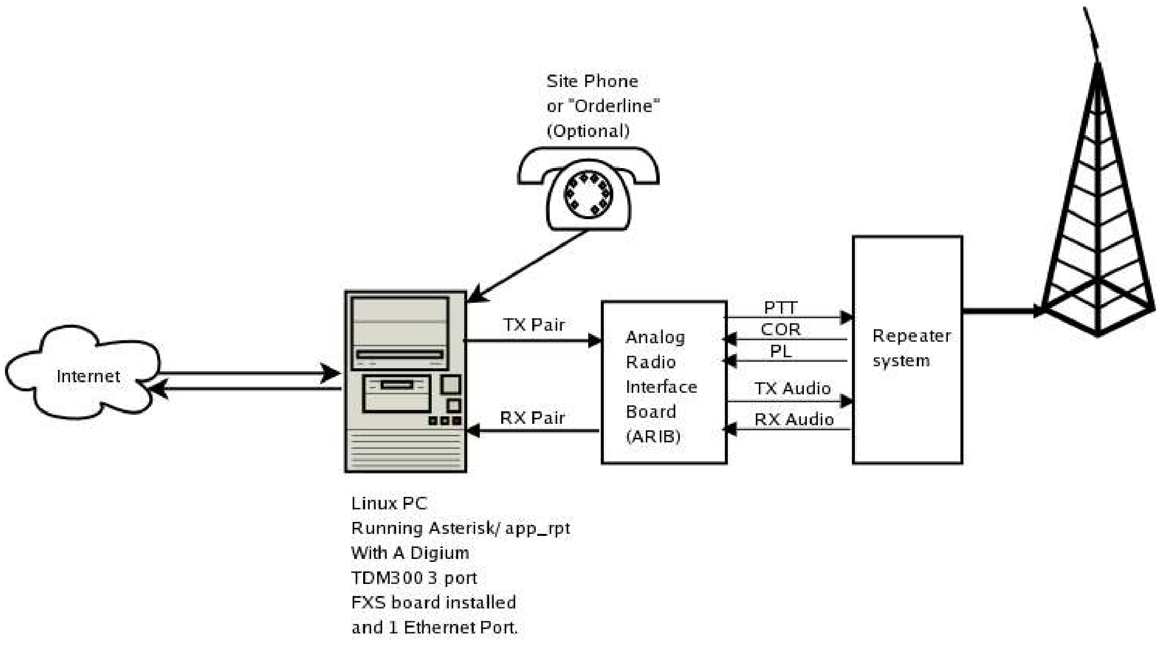

Figure 1: Simple Single Repeater System

I

When we configure Asterisk for use as a repeater controller, we set up the configuration files zaptel.conf and zapata.conf with the type of signaling to be used by a given channel. For the RX channel or RX pair, we use loop start signaling. For the TX channel or TX pair we use ground start signaling. The reasons for this are:

When Asterisk wants to key the repeater transmitter, it places battery on the TX pair. This is sensed by the ARIB which results in the repeater transmitter PTT being activated. Ground start signaling fulfills this need, as the FXS interface in the PC is the only side capable of providing battery.

When the repeater receiver senses a signal, and activates the COR and/or PL signals, the ARIB closes the loop on the RX pair. This causes a current to flow in the RX-pair which is sensed by the FXS interface as an off-hook condition. The off hook condition is then passed to the Asterisk repeater application where it is handled and processed.

Whenever there is a DC current flowing in the TX and RX pairs, AC audio is usually riding on top of this DC current. This audio is picked off of the TX pair and inserted on the RX pair by the ARIB using transformer isolation. The ARIB also contains op-amps and level pots to tailor the transmit and receive audio levels. Optional de-emphasis is available for the receive audio path.

Multiple Repeaters at the Same Site

Running multiple repeaters off the one PC at the same physical site is what our solution excels at. Because our solution can offer a high number of audio ports in a relatively small space, and the PC is capable of handling all of these audio streams in digital form at the same time, each node/repeater combination can be treated as a either a standalone or may be linked to any other node available either locally at the site, or remotely over the Internet. Each node/repeater may have a custom DTMF function list so that some repeaters can have a restricted set of commands available, while other repeaters at the same site can have the full set of commands available. For an example of a typical multi-node site, see figure 2 below

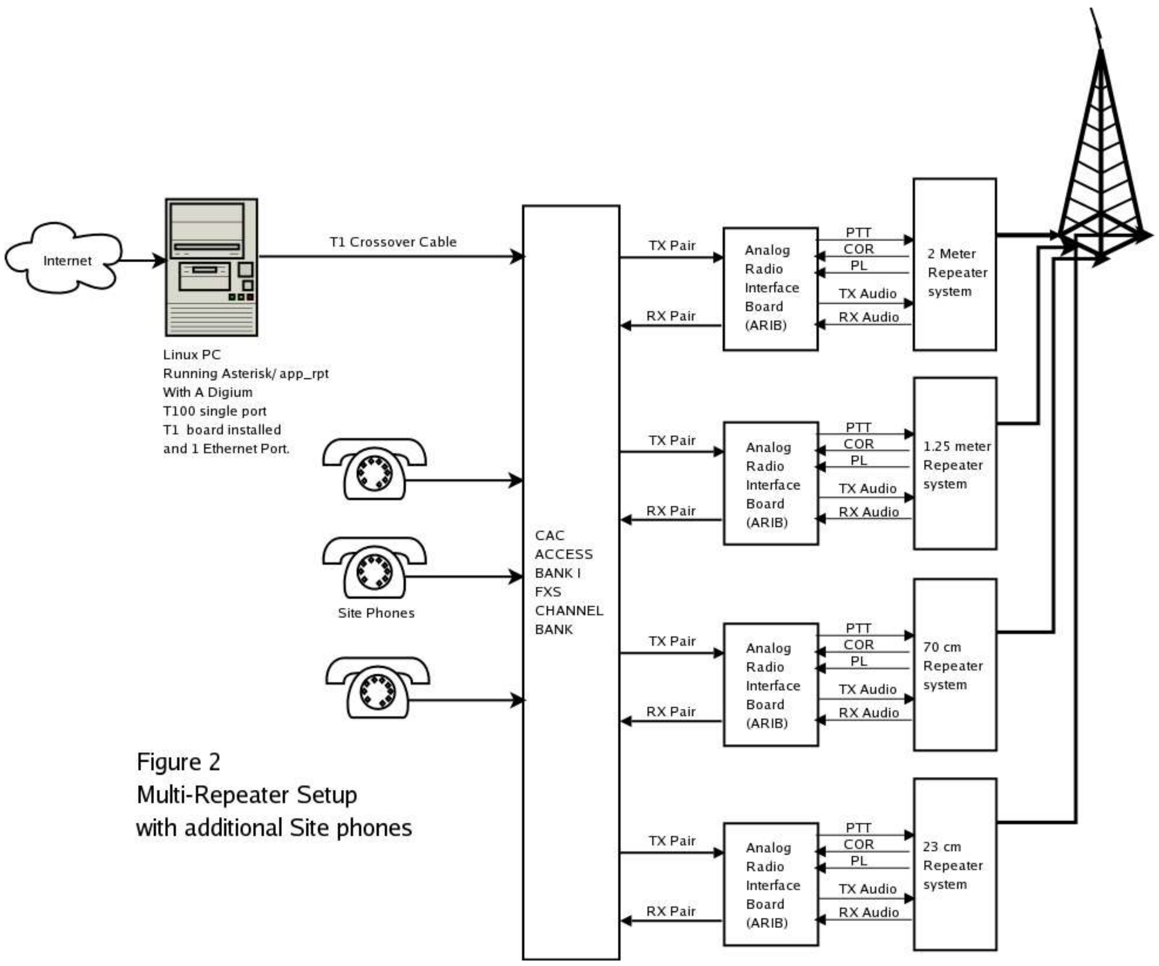

Figure 2
Multi-Repeater Setup
with additional Site phones

Networking Multiple Asterisk Repeater Systems

Figure 3 below shows a hypothetical network of app_rpt Asterisk repeater sites.

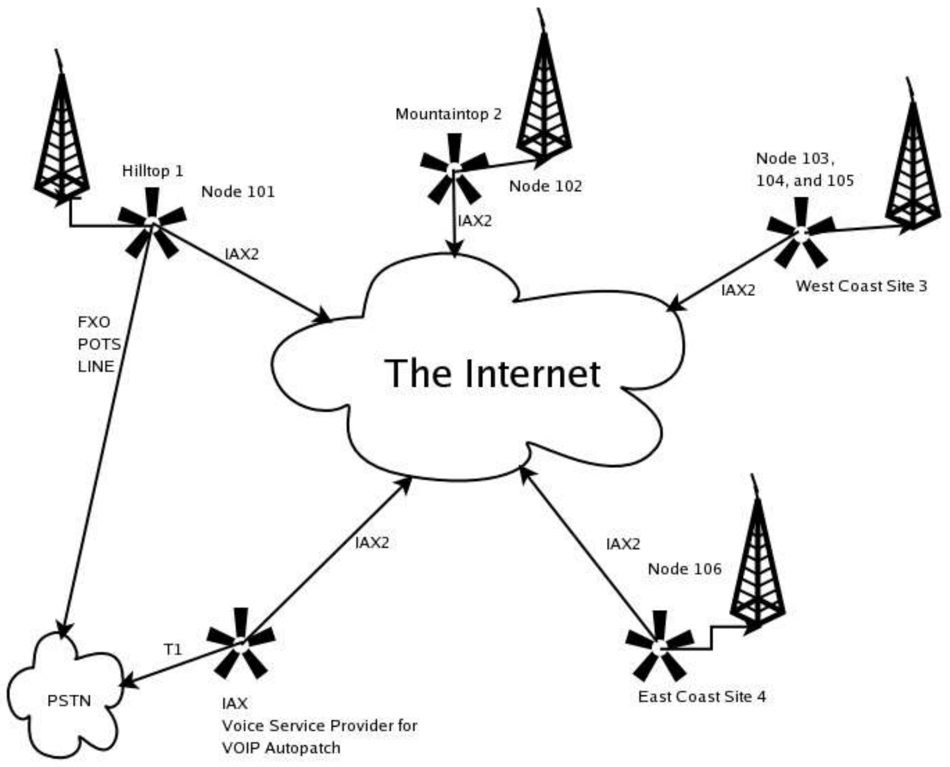

Figure 3 Example Linked System with VOIP Autopatch Capability

The figure shows four repeater sites, plus a fifth site all connected to the Internet. The repeater sites can be anywhere there is broadband Internet service and decent height above average terrain. One of these sites (node 101) has a POTS (plain old telephone service) line connected to it using an FXO (foreign exchange office) interface. FXO interfaces look like telephone sets. Node 101 uses the POTS line for its autopatch. The remaining 3 sites also have an autopatch, but there is no physical phone line present at these sites. These sites get PSTN (public switched telephone network) connectivity from the fifth site which is a VSP (Voice Service Provider).

The IAX2 protocol is used to send audio and control information from one physical site to another. IAX2 is unique in that it is NAT-friendly, and it is bandwidth-efficient. For further information on IAX2, the white paper at http://www.cornfed.com/iax.pdf will go throgh the details. If you want to have a router/hardware firewall between your Asterisk Box and the Internet, the only port you need to open up is port 4569 for TCP and UDP.

We use the GSM codec in Asterisk to compress the signed linear (SLIN) audio from the Zaptel interface before it is inserted into the IAX2 and TCP protocol headers respectively. Using this scheme, the bandwidth overhead per link is approximately 35 kilobits per second per direction. For example, if you have a 1 megabit download speed and 256 kilobit upload speed from your ISP, you should theoretically be able to have up to 7 IAX2 packet streams up and running simultaneously at your site.

Unlike RF-linked systems, any node is capable of connecting to any other node directly without going through any intervening sites. This leaves your other sites free to carry on separate QSO's. Like RF-linked systems with several links terminating at one physical site, there may be multiple packet streams coming from and going to remote sites which terminate in a node (or nodes) at our local site.

Dynamic IP Support

We now support nodes on Internet connections with Dynamic IP addresses. The latest version of app_rpt has been changed to do IP address checking internally. Previous versions of app_rpt relied on permit/deny statements in iax.conf. This limited us to static IP addresses.

The latest version of app_rpt uses symbolic names which are looked up by DNS. If you have Internet service with a dynamic IP address, you would use one of the Dynamic DNS service providers to keep your dynamic IP address current.

When someone requests a connection to your node, app_rpt knows what the symbolic address of the connection should be, the IP number of the originating node,, and the node number of the originating node. With this information app_rpt can do a forward DNS lookup on the symbolic address and compare the two IP addresses. If the IP addresses match, the connection request will be honored.

Additional authentication can be added to iax.conf to require either an MD5 secret, or a public key for authentication.

DTMF Function Decoding

The app_rpt Asterisk application provides a flexible way to define DTMF command sequences. DTMF command sequences are completely user-definable with the only restriction that all DTMF commands begin with '*'. app_rpt allows a separate command table to be defined for every repeater node, however, you can share a common command table between several co-located repeater nodes as well. There can also be a command table defined for the link interface to limit commands issued by a remote system to a subset of the full command set. If a remote base is implemented, there is another separate command table for it as well. The DTMF command tables are defined in rpt.conf.

The execution of a DTMF command is not tied to an unkey event. the function decoder will wait until it receives enough digits which match a known command, then execute that command. If there aren't enough digits to match a command, then the function decoder will wait for the digit timer to expire before re-initializing. This works differently than most repeater controllers out there, but the benefit of this is having extreme flexibility in defining command sequences.

Unlike the other Internet linking solutions out there, our solution allows DTMF commands to be sent from our local site to a designated remote site. This is done by placing your local node in “command mode”. When you go into command mode, your DTMF command are not interpreted locally, they are instead forwarded using a control channel on the IAX2 packet stream to the remote site. When you are finished functioning the remote site, you exit command mode and now all further commands will be interpreted locally.

Telemetry

Telemetry is used to indicate the state of the repeater system. Telemetry consists of voice messages, tones, and morse code strings. Each node can have its own telemetry stanza (group of settings) , or a single telemetry stanza may be common to all nodes.

Courtesy tones may be single or dual tone with no limit on the number of segments. There are a number of courtesy tones pre-defined in rpt.conf. You can customize these or add your own.

Morse strings are useful in those cases where you don't want voice response. They are also used when the talkover ID feature is invoked. ID strings are a special version of morse strings which are not interruptible when a signal appears on the repeater receiver or link. Morse speed, pitch, and level are separately adjustable for ID strings and morse strings.

Adding a Remote Base

app_rpt supports remote bases nodes in addition to repeater nodes. Remote base nodes only require a 2-wire connection (one FXS port) unlike repeater nodes which require a 4-wire connection (2 FXS ports). Our remote base implementation is *autonomous* which means that it can be used independently from a repeater node at the same physical site. In other words, if there is a repeater and a remote base node co-located at the same site, two different users could be linked to the remote base node and the repeater node, and their audio paths will be separate.

The remote base implementation in app_rpt currently supports both fixed-frequency remote bases, as well as frequency-agile remote bases. The hardware required to implement a fixed-frequency remote base consists of an ARIB board, and your remote base radio. If you want to implement a frequency-agile remote base, then an ARIB, a Doug Hall RBC1 adapter, and Kenwood TM-series radios are required.

Autopatch and Telephone Line Control

Autopatching is supported both to and from the radio system.

Radio users may place calls to VOIP endpoints or to numbers on the Public Switched Telephone Network (If you have the appropriate termination services).

There are two reverse Autopatch modes supported: Smart and Dumb:

> Smart mode uses Asterisk's call parking facility and pages radio users to pick up the call. If no one on the radio side picks up the call within a time out period, the caller will be notified and disconnected.
>
> Dumb mode allows a user connects the user directly to the radio system and keeps the repeater node transmitter keyed for the duration of the call.

There is also support for telephone line control of the repeater and remote base nodes. Control operators or supervisors may call in to the system and monitor radio traffic, and issue DTMF commands. The control operator or supervisor may also join in the radio conversation if necessary be special commands to key and unkey the transmitter.

Soft Dispatch Console

app_rpt supports a soft dispatch console called iaxrpt which runs on a Windows or Linux PC. This allows the user to use a computer to talk to radio users in the field using just a PC with a sound card and a microphone/headset. There is a web site for iaxrpt at iaxrpt.qrvc.com.

The Allstar Link Node Address Assignment Authority

The developers maintain a node numbering authority for open and semi-open systems which want to link with each other. Any app_rpt user can request a node assignment from the Allstar Link Node Assignment authority at www.allstarlink.org

Hardware Sources

The URL to purchase assembled and tested boards is www.qrvc.com/radiocards.html.

Digium makes and is the sole source for the TDMx0 PCI interface cards, and the TE110P single-port T1 interface card. Their URL is www.digium.com

CAC FXS Access bank I channel banks can be found used on www.ebay.com. When buying these channel banks make sure you get an FXS firmware with the channel bank and not an SLC96 or TR08 firmware. The SLC96 and TR08 firmware versions will be useless, and it will cost you more to change the firmware than to buy a whole new channel bank.