IAX_Bridge

What is it?

IAX_Bridge (IAXB) is a specially designed bridge that facilitates communication between DVSwitch Server and the AllStar network. Its primary function is to enable seamless connectivity between these two systems.

Why would we consider building something that seems to already exist?

- Performance
  - Size vs AllStar
    - AllStar is about 5 MB
    - IAX_Bridge is about 650 KB
  - CPU usage and memory vs AllStar (RPI 3B+)
    - asterisk: %CPU = 8.2 %Memory = 5.9
    - IAX_Bridge: %CPU = 1.0 %Memory = 1.3
- Complexity
  - Single small configuration file
  - No kernel dependencies (**No DAHDI**)
    - Let us say that again **NO DAHDI**
  - Easy, painless upgrades
    - Simple apt upgrade
- Purpose built
  - Main objectives are bridging and user terminal
  - Better integration with DVSwitch Server
  - Seamless integration with other AllStar nodes
    - Can connect to and receive connection from any AllStar node
    - Uses the same authentication scheme as Allstar
  - Smaller security attack surface
- Transportability
  - Linux and Mac. Maybe even Windows
  - Not distribution dependant (Raspian, Armbian, Chrome OS, x86, x64)
  - Got a toaster that runs Linux? We probably run there too..
- Connectivity
  - USRP
    - Used to communicate with DVSwitch components.
  - IAX2
    - Used to communicate with AllStar nodes.

To be fair, what is IAXB NOT?

- IAXB is not a general purpose Allstar replacement
  - See Purpose built above
- IAXB is not primarily for use with RF hardware (with caveats today)
  - Will work with MMDVM modem in FM Networking mode (USRP)
- IAXB is not supported by Allmon/Supermon or any AMI based dashboard
  - The DVSwitch hUC program gives you full control of the node as well as provide a web browser based transceiver with control over who uses it

IAXB is not compatible with Echolink, TLB, IRLP

These type of connections would be addressed by a purpose built bridge.

IAXB does not speak

IAXB is not a repeater controller

IAXB is not a AllStar hub (TBD)

Setup

IAX_Bridge is designed to be simple to install and configure. Unlike Asterisk there is a single executable and a single configuration file. A simple bridge (to a digital network) or an interactive node (with Analog_Reflector) are the two most common use cases.

Installation

Package based install

Apt install iax-bridge (not yet available)

Edit the ini file to match your specific information

Manual install

Copy/rename the binary into the /opt/IAX_Bridge directory

Copy the IAX_Bridge.ini file into the /opt/IAX_Bridge directory

Copy/create a system unit file and install in the /lib/systemd/system directory

Edit the ini file to match your specific information

INI file setup

[IAX]

call	*The node callsign*
node	*The network node number*
regServerAddress	*The server used to register with the network.*
password	*The password used to register with the network*
startNode	*Optional node to connect to at startup*
usrpAddress	*The address of the USRP application*
rXPort	*UDP port to receive from the USRP application (import)*
tXPort	*UDP port to transmit to the USRP application (export)*
displayLevel	*What type of info should IAXB display to the console*
logLevel	*What type of info should IAXB write to a log file*
logFilePath	*Where to place the log file*
iaxPort	*UDP port used to communicate with the network*
statsEnabled	*Send connection statistics to ASL*
keyQueryEnabled	*Requests to see who is transmitting (Broken by HamVoIP)*
timeoutSeconds	*Transmission timeout (both directions)*

[nodes]

Nodes you want to connect to that are not part of the ASL network

Can be used to block unwanted AllStar nodes

[MACROS]

System event macros

OnConnect

run program when a connection to another node is established

OnDisconnect

run program when a connection to another node is released

OnRejected

Run program when an incoming or outgoing connection fails

OnTelemetry

Variables depend on the telemetry received

User defined macros

Execution

System units

Systemctl {start | stop | status} iax_bridge

Command line

./IAX_Bridge IAX_Bridge.ini

The log file

Where is it?

Location is defined in IAX_Bridge.ini by *logFilePath*

Default location /var/log/dvswitch/IAX_Bridge.log

How to use the log file to debug an issue

tail -f /var/log/dvswitch/IAX_Bridge.log

Adjust the logLevel and displayLevel

Connection issues

Are you registered?

IAXC_EVENT_REGISTRATION

Connection Timeout

Call disconnected by remote

Examples

A simple digital to IAX bridge

IAXB < —- > AB < — > MB

Web based IAX node

IAXB < — > AR

A stand alone IAX node

IAXB < — > pyUC

IAXB < — > DVSM? (this make almost no sense)

FAQ

Connection rejected?

Just like a AllStar node, your bridge registers with the ASL registration server

It takes a few minutes for the updated node list to be sent out to all the AllStar

Nodes to receive the updated node list.

One way audio

Metadata not correct?

Blacklisting nodes

To block inbound and outbound connections for node 1234

Add this to your nodes section of IAX_Bridge,ini

1234 = radio@127.0.0.1:4569/1234,NONE ; block node 1234

Incoming connections fail

Unsupported codec?

Adding private nodes

You can connect to and receive calls from nodes outside of the network

Add this to the nodes section of IAX_Bridge,ini

1579 = radio@1111.2222.3333.4444:4569/1579,NONE ; add node 1579

Supported codecs?

ULAW, ALAW

Macros

IAX_Bridge can execute external commands or scripts when system events occur or user-defined macros are invoked. System events include OnConnect, OnRejected, OnDisconnect, and OnTelemetry. User-defined macros start with a "*" and match a macro name specified in the [MACROS] section of the INI file. Shell environment variables like IAX_CONFIG_PATH and IAX_CONNECTION can be utilized, and macros invoke a shell with the constructed command line.