A batman-adv mesh network over AX.25

Setup a mesh network with any transceiver imaginable

Introduction

The concept

asdf

asdf

Setting up a batman-adv mesh network over the AX.25 protocol and thus every transceiver that supports voice communication (including cheap PMR446 transceivers, shortwave transceivers for long range, aux cables for testing and many, many more) or that can be connected to a TNC (like the popular CC1101 IC using a FSK TNC) is a project that I have been working on for a long time now and that I can imagine many people would really appreciate. In this document I will describe how to first make a point-to-point connection using batman-adv over AX.25 as a first step towards this goal and then setup the mesh network using an example setup.

+ Mesh Network Introduction
+ DHCP
+ MOSANET/MOSALINK Usage
+ Rechtliches
+ Fallback solution for other modes
+ Speed
+ No actual modem etc. needed
+ RPI/BBB
+ Revision and las
+ Alpha Horizon document template
+ Install everything simultaneously
+ Open-Source
+ TNC Reference designs and complete online website
+ Alpha Horizon Website link
- 2017-06-25 15:36 Felix Pojtinger

The setup
asdf
(Setup diagram here)

The possibilities
asdf

+   Exciting usage examples

Table of contents

Requirements

In order to build follow this guide, there are some requirements that have to be fulfilled before you can start.

Knowledge

asdf
+   Linux Experience

Hardware

asdfasdf
+   Testing system

Government regulations

asdfasdf
+   Not a ham?

Computers

asdfasdf

Transceivers

asdfasdf

Soundmodem or other TNCs

asdfasf

Software

asdfasdfasdf

+ Testing system
+ Tested Linux Systems
+ batman-adv versions

Layer 1 (electrical connections) setup

This section will describe how to setup the electrical connections from for example your computer to the transceiver. This is obviously different on every computer so this is just a general overview.

PC to soundmodem TNC or other TNC

asdfasdfasdf

Soundmodem TNC or other TNC to transceiver

asdfasdf

Transceiver to antenna

asdfasdf

Example testing setup

asdfasdfasdfasdf

Layer 2 (AX.25 via the soundmodem kernel module)

This section will describe how to install and setup the soundmodem kernel module to provide a AX.25 Interface.

Overview

- What is AX.25?
- What is a kernel module?

Installation

soundmodem - Install:

sudo apt update
sudo apt upgrade
sudo apt install soundmodem libax25 ax25-apps ax25mail-utils ax25-tools ax25-xtools

Setup

Soundcards - Show:

cat /proc/asound/cards

soundmodem - Configure (Terminal #1):

sudo soundmodemconfig

soundmodem - Configure (GUI):

Head over to https://www.george-smart.co.uk/aprs/ax25_soundmodem/ and follow the wonderful tutorial by M1GEO on how to configure soundmodem via the GUI. Make sure that "mode" in IO of the channel config is set to "alsa" and use the sound card name from above as the ALSA Audio Driver, its always plughw:x,0 where x is the number you got from the above command! Do not use "default"! Setup the TNC as MKISS TNC as use either afsk or fsk, both will work (fsk up to 38400 Baud, however careful audio volume tuning is needed! Also call the interface ax0 instead of sm0. Note down your interface name (ax0), your callsign and baudrate. Choose a different ip (change the last digit) for every node!

soundmodem - Configure (Terminal #1):

This is also described by M1GEO, however I wanted to include a light version here.

sudo nano /etc/ax25/axports

Copy one of the other lines of the already existing interfaces (without the #) and modify it by adding the information (ax0, callsign, baudrate) you've noted down before.

Unwanted processes - Stop:

sudo service smbd stop
sudo service nmbd stop

Ethernet emulation layer (eoax kernel module)

This section will describe how to install and setup the eoax kernel module to provide Ethernet functionality to the ax0 interface.

Overview

- Why do I need this?
- Ethernet vs AX.25

Installation

Git - Install

```
sudo apt install git
```

eoax Github repository - Clone

```
git clone https://github.com/chazapis/eoax
```

sources.list - Edit

```
cd ~/
sudo nano /etc/apt/sources.list
```

Using nano uncomment `deb-src http://de.archive.ubuntu.com/ubuntu/ zesty main restricted` by removing the `#` before it, save and then close nano

Linux kernel build tools and kernel source - Install

```
sudo apt update
sudo apt-get build-dep linux-image-$(uname -r)
sudo apt-get source linux-image-$(uname -r)
```

Linux kernel - Patch

Look up the name of the Linux kernel source folder you've just downloaded and cd there:
```
ls
cd linux-4.10.0

patch -p0 < ~/eoax/patches/ax25_ui_type-3.19
sudo cp include/net/ax25.h /usr/src/linux-headers-$(uname
-r)/include/net/
cd net/ax25
```

Kernel module - Compile #1

```
make -C /lib/modules/$(uname -r)/build M=$(pwd) modules
```

Kernel module - Sign

Kernel module signing is required since Linux kernel 3.7.
```
cd /usr/src/linux-headers-$(uname -r)/certs
sudo touch x509.genkey
sudo nano x509.genkey
```

Using nano paste the following, save and the close nano:
```
[ req ]
default_bits = 4096
distinguished_name = req_distinguished_name
```

```
prompt = no
string_mask = utf8only
x509_extensions = myexts

[ req_distinguished_name ]
CN = Modules

[ myexts ]
basicConstraints=critical,CA:FALSE
keyUsage=digitalSignature
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid
```

Continue with signing the module:
```
cd /usr/src/linux-headers-$(uname -r)/certs
sudo openssl req -new -nodes -utf8 -sha512 -days 36500 -batch -x509
-config x509.genkey -outform DER -out signing_key.x509 -keyout
signing_key.pem

sudo /usr/src/linux-headers-$(uname -r)/scripts/sign-file sha512
/usr/src/linux-headers-$(uname -r)/certs/signing_key.pem
/usr/src/linux-headers-$(uname -r)/certs/signing_key.x509
~/linux-4.10.0/net/ax25/ax25.ko
```

Kernel Module - Compile #2
```
cd ~/linux-4.10.0/net/ax25
sudo make -C /lib/modules/$(uname -r)/build M=$(pwd) modules_install
sudo su
echo "override ax25 * extra" >> /etc/depmod.d/ubuntu.conf
exit
sudo depmod
cp Module.symvers ~/eoax/drivers/net/hamradio/
cd ~/eoax
make
```

Layer 2(.5) (batman-adv kernel module)

This section will describe how to install and setup the batman-adv kernel module and use DHCP to automatically assign IP addresses.

Overview
- What is batman-adv?
- Why not just use Ethernet?
- Why to use DHCP?
- Limitations (scalability) and solutions

Installation

batman-adv - Install:

sudo apt update
sudo apt upgrade
sudo apt install batctl bridge-utils openssh-server openssh-client

Setup

batman-adv - Setup:

sudo modprobe batman-adv
sudo batctl if add wlan0
sudo ip link set up bat0

Startup

This section will describe how to activate each interface and automatically assign ip addresses to every interface.

Activating soundmodem

sudo soundmodem
sudo service smbd stop
sudo service nmbd stop

Activating eoax

sudo modprobe eoax
sudo ip link set dev eoax0 down
sudo ip link set dev eoax0 up
sudo avahi-autoipd eoax0 -D
ip a

Activating batman-adv

sudo modprobe batman-adv
sudo batctl if add eoax0
sudo ip link set dev bat0 down
sudo ip link set dev bat0 up
sudo avahi-autoipd bat0 -D
ip a

Testing and usage examples

This section will describe how to test and use the setup as a whole and show some examples of its usage.

Overview and general approach ideas

asdfasdfasdf

Useful command collection

asdfasdf
+ From sections above

Using Ping, FTP, Apache, SSH etc.

asdfasdf

Autostart

asdfasdf

The future

asdf

Legal notes

asdf
+ Standard Alpha Horizon Document Template Information