Building a Better Demodulator for APRS / AX.25 Packet Radio

Part 1, 1200 Baud AFSK

John Langner, WB2OSZ

April 2015

These are not professionals on a closed course.

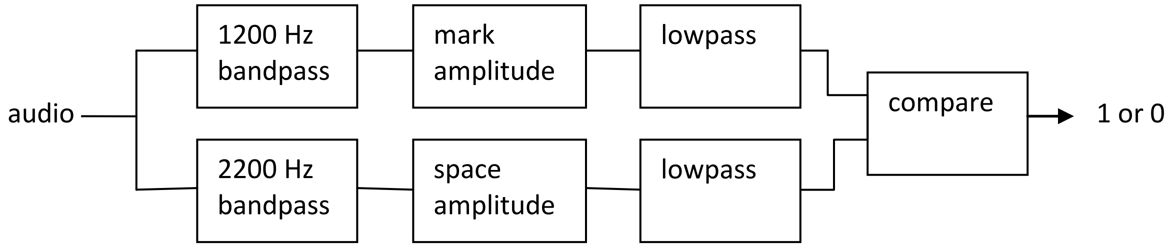
You CAN try this at home.

Sometimes it's a little mystifying why an APRS / AX.25 Packet TNC will decode some signals and not others. A weak signal, buried in static, might be fine while a nice strong clean sounding signal is not decoded. Here we will take a brief look at what could cause this perplexing situation and a couple things that can be done about it.

Demodulator Basics

The most common method for transmission of AX.25 Packet Radio is based on the old Bell 202 standard. Data is sent at 1200 bits per second using two different audio frequencies: 1200 & 2200 Hz, often called "mark" and "space," to represent the logic 1 and 0 values. This is called audio frequency shift keying (AFSK). A demodulator needs to listen for those two tones and decide which one is stronger.

Here is the general idea for one type of demodulator:



We start off with a bandpass filter for each of the audio frequencies. In the old days, you might use an inductor and capacitor. Later we progressed to active filters with op amps. Today it can be done in software.

What we are interested in is the amplitude of the tone, not the sine wave. In the old days, this would be a full wave rectifier. The software equivalent is taking the absolute value so the negative peaks become positive.

A low pass filter allows the lower speed data through while removing the higher frequency ripple and noise.

Finally, we have a comparator to decide which tone is stronger and the result is logic 1 or 0.   This goes to an HLDC decoder that looks for patterns in the bit stream and spits out groups called “frames.”

This type of demodulator is not that hard to implement and works well for ideal signals. Unfortunately, we don’t hear too many ideal signals, so it becomes a little more challenging. The problem is that we are sending digital data through transceivers designed for voice. The audio circuits designed to improve the intelligibility of the human voice distort the data and make it less likely to be decoded properly.

A Look at Ideal Signals

First let’s start off looking at the characteristics of an idea signal. We will generate some ideal frames and decode them. You might want to follow along at home and try some variations on these. You will need Dire Wolf version 1.2, which is still under development when this is being written. This is still evolving so the exact appearance and numbers could change, but the principles illustrated remain the same.

```
gen_packets -o test1.wav
atest test1.wav

DECODED[1] WB2OSZ-15 audio level = 49(50/51)
[0] WB2OSZ-15>TEST:,The quick brown fox jumps over the lazy dog!  1 of 4

DECODED[2] WB2OSZ-15 audio level = 49(51/51)
[0] WB2OSZ-15>TEST:,The quick brown fox jumps over the lazy dog!  2 of 4

DECODED[3] WB2OSZ-15 audio level = 49(51/50)
[0] WB2OSZ-15>TEST:,The quick brown fox jumps over the lazy dog!  3 of 4

DECODED[4] WB2OSZ-15 audio level = 49(51/51)
[0] WB2OSZ-15>TEST:,The quick brown fox jumps over the lazy dog!  4 of 4
4 packets decoded in 0 seconds.
```

“gen_packets” generates some AX.25 frames, converts them to audio, and puts the result in the specified file.   “atest” reads the audio file and extracts the frames.

In each case, we see an audio level composed of 3 parts:

- Received audio signal level.
- Strength of the mark tone.

- Strength of the space tone.

Notice how they are all quite close.

Add Some Noise

Next we will generate 100 frames with increasing levels of random noise and try to decode them. Most of the resulting output was omitted to save space.

```
gen_packets -n 100 -o test2.wav
atest test2.wav

DECODED[1] WB2OSZ-15 audio level = 50(51/52)
[0] WB2OSZ-15>TEST:,The quick brown fox jumps over the lazy dog!  0001 of 0100

DECODED[2] WB2OSZ-15 audio level = 50(51/51)
[0] WB2OSZ-15>TEST:,The quick brown fox jumps over the lazy dog!  0002 of 0100

…

DECODED[26] WB2OSZ-15 audio level = 72(57/53)
[0] WB2OSZ-15>TEST:,The quick brown fox jumps over the lazy dog!  0026 of 0100

DECODED[27] WB2OSZ-15 audio level = 75(56/59)
[0] WB2OSZ-15>TEST:,The quick brown fox jumps over the lazy dog!  0027 of 0100

…

DECODED[61] WB2OSZ-15 audio level = 109(68/61)
[0] WB2OSZ-15>TEST:,The quick brown fox jumps over the lazy dog!  0061 of 0100

DECODED[62] WB2OSZ-15 audio level = 113(60/74)
[0] WB2OSZ-15>TEST:,The quick brown fox jumps over the lazy dog!  0064 of 0100

…

64 packets decoded in 1 seconds.
```

Notice how it starts off fine with little noise. All 3 audio levels are about the same. As the noise level increases, the overall audio input level goes up to more than double the original number. The mark and space filter outputs increase a little, because some of the wide-band noise gets through, and we notice larger inconsistencies between the levels. One or the other might be higher depending on the random noise. Eventually, we reach the point where the noise level is so high the frames can't be decoded.

When the first audio level number is significantly larger than the others, it indicates we have a poor signal to noise ratio.

The Real World

There is another reason why we see imbalances between the two tones even when no noise is present. You should find a very good explanation here:

http://www.febo.com/packet/layer-one/transmit.html

Go read it now. I'll be waiting here, to continue, when you get back…..

The important points are,

- Microphone audio is "pre-emphasized" to boost the higher frequencies before transmission.
- Speaker audio is "de-emphasized" to attenuate the higher frequencies. In theory these should cancel out.
- When the transmit audio level is set too high, limiting, which is applied after the pre-emphasis, causes a greater decrease in the higher (space) tone.

It gets worse.

- Some systems use a flat bandpass on transmit rather than boosting the higher frequencies.
- Some systems use a flat bandpass on receive rather than attenuating higher frequencies.
- When the sender and receiver use different conventions, the imbalance is even greater.

Playing with simulated signals can get us only so far. Let's look at some actual signals. WA8LMF did the ham radio community a great service by providing a collection of real-world imperfect signals that can be used for testing. Learn about it and download a copy from here: http://wa8lmf.net/TNCtest/

- Track 1 has a flat response on receive.
- Track 2 has the same thing with de-emphasis typical of many receivers. This has become the de facto standard for measuring TNC receive performance. Google for "tnc test cd" and you will find scattered test results.

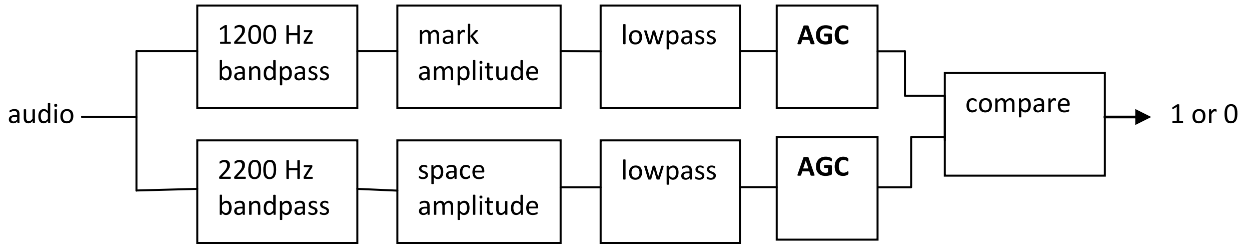
When we examine the ratio of mark / space amplitude, we find:

- Track 1: 0.53 to 1.38 with a median of 0.81.
- Track 2: 1.73 to 3.81 with a median of 2.48.
- Heard locally (from speaker): 1.48 to 3.41 with a median of 2.70.

My first naïve attempt at building a demodulator worked great on ideal signals but performance was very disappointing with real-world signals where there were considerable variations between the strengths of the mark and space tones.

Earlier Solution

My solution was to add an automatic gain control (AGC) for the mark and space amplitudes. This normalized both signals to the same level before making the comparison.



This works well, decoding 970 frames from Track 2 of the TNC Test CD. The “-P C” option, below, means use the default decoder parameters from version 1.1.

```
atest -P C 02_Track_2.wav

DECODED[1] Digipeater N6EX-5 audio level = 9(6/3)
[0] WA6YLB-4>APRS,N6EX-5*:$ULTW00000000----0000----000086A00001----
0000000000000000<0x0d><0x0a>

DECODED[2] KD6FVP-2 audio level = 19(13/5)
[0] KD6FVP-2>APS224,N6EX-1,WIDE1:>152343z[224]*We know most of your
faults!!!<0x0d>

…

DECODED[969] Digipeater N6EX-4 audio level = 13(5/3)
[0] N6XQY-12>GPSLJ,N6EX-
4*:$GPRMC,021841.28,A,3346.1449,N,11755.4088,W,00.0,000.0,231105,13.,E*71<0x0d>

DECODED[970] Digipeater N6EX-4 audio level = 12(5/3)
[0] KN6DB-14>GPSLK,N6EX-
4*:$GPRMC,021851,A,3348.8470,N,11800.1685,W,000.0,274.0,231105,013.4,E*6D<0x0d>
<0x0a>

970 packets decoded in 54 seconds.
```

The “E” decoder, new in version 1.2, adds a bandpass filter between the radio and the mark/space filters. This helps with noisy signals and gives us about another 18 frames. These numbers might vary a little as the application evolves.

```
atest -P E 02_Track_2.wav

DECODED[1] Digipeater N6EX-5 audio level = 9(6/3)
[0] WA6YLB-4>APRS,N6EX-5*:$ULTW00000000----0000----000086A00001----
0000000000000000<0x0d><0x0a>

DECODED[2] KD6FVP-2 audio level = 19(13/5)
[0] KD6FVP-2>APS224,N6EX-1,WIDE1:>152343z[224]*We know most of your
faults!!!<0x0d>

…
```

```
DECODED[987] Digipeater N6EX-4 audio level = 16(3/1)

[0] N6XQY-12>GPSLJ,N6EX-
4*:$GPRMC,021841.28,A,3346.1449,N,11755.4088,W,00.0,000.0,231105,13.,E*71<0x0d>

DECODED[988] Digipeater N6EX-4 audio level = 16(2/1)

[0] KN6DB-14>GPSLK,N6EX-
4*:$GPRMC,021851,A,3348.8470,N,11800.1685,W,000.0,274.0,231105,013.4,E*6D<0x0d>
<0x0a>

988 packets decoded in 63 seconds.
```

Where did that **02_Track_2.wav** file come from? Waiting 25 minutes and 49 seconds for each test would get boring quickly. The tracks were "ripped" from the TNC Test CD and saved in the WAV file format. This allows us to run each test in about a minute.

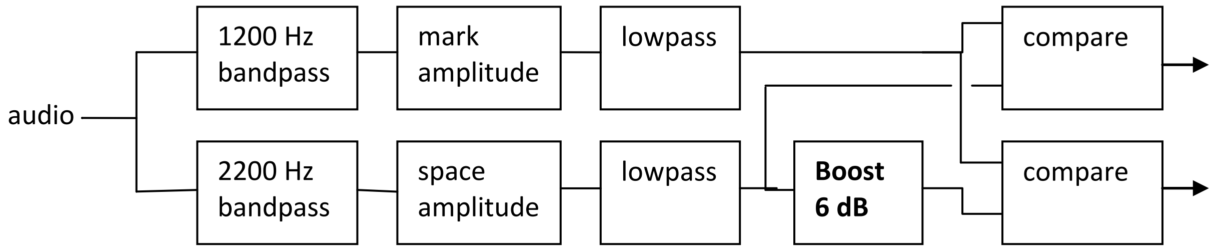
All of the demodulator parameters involve tradeoffs. We want the filters to be narrow to exclude undesired signals (noise) but they must be wide enough to let the desired signals through. The automatic gain control also involves tradeoffs in how quickly it changes gain in response to input signal changes. What might be optimal for one frame might not be as good for another.

Another Approach

4X6IZ had another interesting approach described in QEX:
http://www.tau.ac.il/~stoledo/Bib/Pubs/QEX-JulAug-2012.pdf

Two separate demodulators, tuned in different ways, are run in parallel. One has a pre-filter with flat response in the range of interest. The other boosts the space frequency by 6 dB. One decoder did better on Track 1; the other did better on Track 2. After removing duplicates, decoded by both, up to 966 frames were decoded.

The unfortunate consequence of running two completely separate demodulators is that it requires twice as much CPU power. The filters are very computationally intensive. An amplifier, comparator, and HDLC decoder take much less compute power. Here is little variation on the same idea that does not take significantly more computation.



It doesn't take much more CPU time so why stop at two? I tried a little experiment with 9 different gains for the space tone at the same time.

Track 1 – Receiver with flat response.

Here we use the “E+” decoder which is now the default for Dire Wolf version 1.2. The “+” part means we use multiple gains and slicers rather than the AGC approach.

```
atest -P E+ 01_Track_1.wav

…

DECODED[49] Digipeater W6SCE-10 audio level = 52(15/22)     __|||____
[0] KB6CYS>BEACON,W6SCE-10*:WEATHER STATION ON-LINE<0x0d>

DECODED[50] Digipeater N6EX-5 audio level = 58(18/26)     __|______
[0] KD6LAY>APW277,W6PVG-3,N6EX-5*:=3438.88N/11815.40WyPHG5660/WinAPRS 2.7.7 -
277-<630><0x0d>

…

DECODED[1009] Digipeater W6SCE-10 audio level = 50(15/21)     |||||____
[0] K6SYV-10>ANP391,W6SCE-10*:!3444.00NS12000.40W#PHG7730/Wn,SCAn/FIGUEROA
Mt./A=003248<0x0d>

DECODED[1010] Digipeater N6EX-4 audio level = 51(16/26)     |||||____
[0] N6XQY-12>GPSLJ,N6EX-
4*:$GPRMC,021841.28,A,3346.1449,N,11755.4088,W,00.0,000.0,231105,13.,E*71<0x0d>

DECODED[1011] Digipeater N6EX-4 audio level = 51(16/26)     ||||||___
[0] KN6DB-14>GPSLK,N6EX-
4*:$GPRMC,021851,A,3348.8470,N,11800.1685,W,000.0,274.0,231105,013.4,E*6D<0x0d>
<0x0a>

1011 packets decoded in 68 seconds.
```

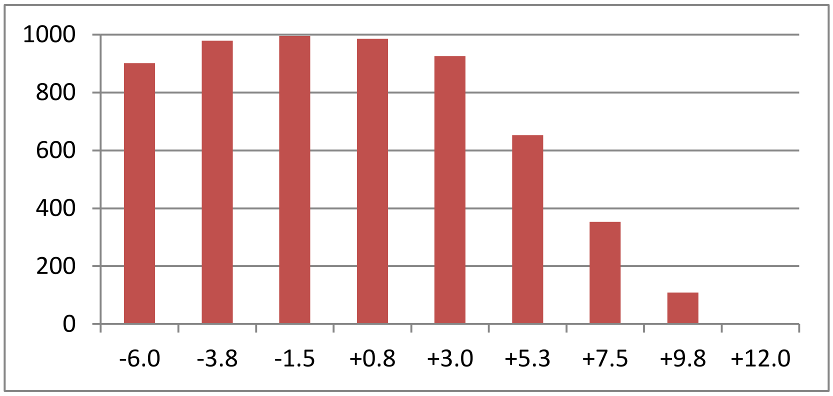
You probably noticed something new: vertical bars and underscores after the audio levels. Each one of these character positions corresponds to a decoder with different gains for the space frequency.

| means a frame was received with a correct CRC.
_ means no frame was received.

As an example, _||______ means the second and third decoders successfully received the frame.

When we add up the number of frames received, for each of the decoders, this is what we find:

Gain dB	Frames
-6.0	901
-3.8	978
-1.5	995
+0.8	985
+3.0	925
+5.3	653
+7.5	353
+9.8	109
+12.0	0



If we had to limit ourselves to a single comparator, the optimum gain would be around -1.5 dB producing 995 frames.

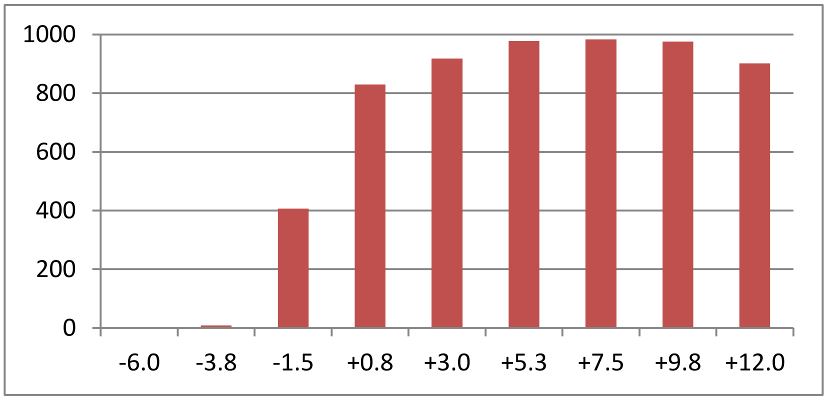
When we run all 9 at the same time and remove duplicates, we get 1011 frames.

Track 2 – Typical de-emphasis found with speaker output.

Now we try the same thing for Track 2.

```
atest -P E+ ../02_Track_2.wav

…

DECODED[49] Digipeater W6SCE-10 audio level = 26(2/1)     __|||||__
[0] KB6CYS>BEACON,W6SCE-10*:WEATHER STATION ON-LINE<0x0d>

DECODED[50] Digipeater N6EX-5 audio level = 7(3/1)       ______|__
[0] KD6LAY>APW277,W6PVG-3,N6EX-5*:=3438.88N/11815.40WyPHG5660/WinAPRS 2.7.7 -
277-<630><0x0d>
…

DECODED[1002] Digipeater W6SCE-10 audio level = 19(2/1)     __|||||||
[0] K6SYV-10>ANP391,W6SCE-10*:!3444.00NS12000.40W#PHG7730/Wn,SCAn/FIGUEROA
Mt./A=003248<0x0d>

DECODED[1003] Digipeater N6EX-4 audio level = 16(3/1)     __||||||_
[0] N6XQY-12>GPSLJ,N6EX-
4*:$GPRMC,021841.28,A,3346.1449,N,11755.4088,W,00.0,000.0,231105,13.,E*71<0x0d>

DECODED[1004] Digipeater N6EX-4 audio level = 16(2/1)     __|||||||
[0] KN6DB-14>GPSLK,N6EX-
4*:$GPRMC,021851,A,3348.8470,N,11800.1685,W,000.0,274.0,231105,013.4,E*6D<0x0d>
<0x0a>

1004 packets decoded in 69 seconds.
```

Gain dB	Frames
-6.0	0
-3.8	9
-1.5	407
+0.8	829
+3.0	918
+5.3	978
+7.5	983
+9.8	975
+12.0	901



Notice how the results are now skewed toward the right. We need to add more gain to compensate for the receiver de-emphasis of the higher frequencies. The same frame that was received by the second and third decoders, on Track 1, is now decoded by the seventh decoder.

If we had to limit ourselves to a single comparator, the optimum gain would be around +7.5 dB producing 983 frames.

When we run all 9 at the same time and remove duplicates, we get 1004 frames.

Summary of Results

What else can we do pull a few more frames out of the noise?

Dire Wolf has a unique feature where it will attempt to fix frames with a bad CRC. This has been available since February 2013 and is described in the **User Guide** section called "*One Bad Apple Don't Spoil the Whole Bunch…*"   Use the "-F 1" option with "atest" to activate the feature.

The final results are:

Gain for balancing tones	Single bit fix-up	Command Line Options	Time, seconds	Track 1 Frames Decoded	Track 2 Frames Decoded
Automatic	no	-P E -F 0	63	993	988
9 different, remove duplicates	no	-P E+ -F 0	69	1011	1004
Automatic	yes	-P E -F 1	63	1012	1001
9 different, remove duplicates	yes	-P E+ -F 1	73	1021	1022

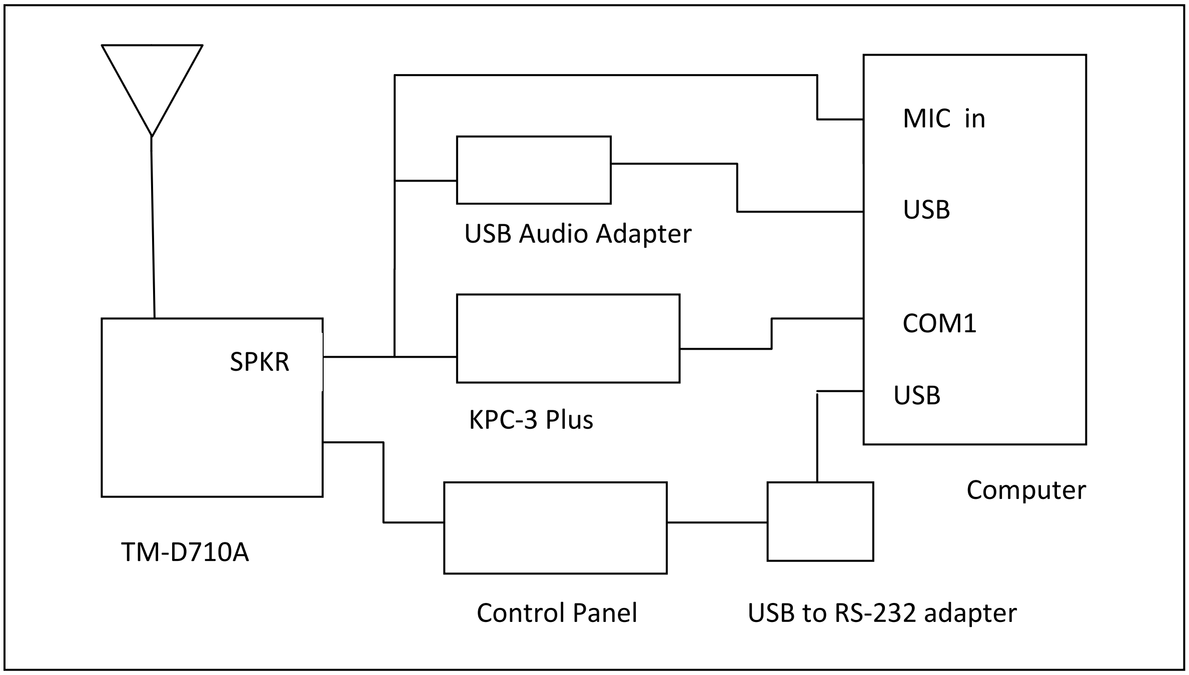
A similar side-by-side real-time real-world comparison first appeared in the **User Guide** for Dire Wolf version 0.9. Let's try it again with improved software version 1.2.

1200 Baud hardware TNC comparison

Here we compare 1200 baud decoder performance against two popular hardware based solutions. This was done early in the version 1.2 development before the new improved "E+" decoder.

For this experiment we need:

- Antenna, outside on the roof.
- A cheap USB Audio Adapter (http://www.adafruit.com/product/1475 - You will be disappointed if you try using this with a Raspberry Pi.)
- Kantronics KPC-3 Plus
- Kenwood TM-D710A
- Serial communication cable for D710A (http://www.amazon.com/gp/product/B000068OER is a lower cost alternative to the official Kenwood PG-5G) – connect to COM port on control panel.
- Audio Y cables, RS232-cables.
- PC running Ubuntu Linux. Connect up everything as shown below.



Prepare KPC-3 Plus

Using some sort of terminal emulator application, such as minicom, connect to /dev/ttyS0. Disable any digipeater settings or beaconing (DIGIPEAT, UITRACE, UIDIGI, UIFLOOD, BEACON, BLT) so it is not distracted by trying to transmit. Beacons also show up like monitored transmissions. Enable monitoring:

```
MONITOR ON
```

You should see received packets being displayed. Exit from the terminal application.

Prepare D710A

Use the TNC button on the control panel to select "PACKET12" (not APRS) mode. Enable the COM port with menu 604.

Using some sort of terminal emulator application, connect to /dev/ttyUSB0. Disable any digipeater settings or beaconing so it is not distracted by trying to transmit. Enable monitoring:

```
MONITOR ON
```

You should see received packets being displayed. Exit from the terminal application.

Prepare Dire Wolf

Be sure to use version 1.2 or later for best performance. In this test we are using Linux so that device name syntax is shown.

Two different configuration files were prepared. The first (direwolf.conf0) will use the default audio input on the motherboard.

```
CHANNEL 0
MODEM 1200 1200 2200 C+
FIX_BITS 0
```

Note that attempted bad bit fix-up is disabled so we count only error-free frames. This provides a fair apples-to-apples comparison against the other systems without this feature.

Prepare a second configuration file (direwolf.conf1) like this.

```
ADEVICE  plughw:2,0
CHANNEL 0
MODEM 1200 1200 2200 C+
AGWPORT 8010
KISSPORT 8011
FIX_BITS 1
```

This provides the more typical usage with the default FIX_BITS value. A $5 external USB Audio Adapter is being used to dispel the rumor that you need an expensive sound card for good results.

Start up two different Dire Wolf instances, with different configuration files, in different windows.

```
direwolf -c direwolf.conf0

direwolf -c direwolf.conf1
```

Compare them.

Run the “aclients” test fixture with command line arguments like this

```
aclients /dev/ttyS0=KPC3+ /dev/ttyUSB0=D710A 8000=DireWolf-0 8010=DireWolf-1
```

Each command line argument is a serial port name or a TCP port number. Notice how we use two different port numbers for the two instances of Dire Wolf. The part after “=” is just a comment to label the results.

Packets are collected from 4 different sources and printed side-by-side in columns for each TNC. A gap means that TNC did not decode the frame that others did.

It starts off looking like this with the first couple packets being received by everybody.

```
john@hamshack:~$ aclients  /dev/ttyS0=KPC3+  /dev/ttyUSB0=D710A  8000=DireWolf-0 8010=DireWolf-1
Client 3 now connected to DireWolf-1 on localhost (127.0.0.1), port 8010
Client 2 now connected to DireWolf-0 on localhost (127.0.0.1), port 8000
Client 0 now connected to KPC3+ on /dev/ttyS0
Client 1 now connected to D710A on /dev/ttyUSB0
W1OEM-4>ID,EKONCT,W1MRA*,WI   W1OEM-4>ID,EKONCT,W1MRA*,WI   W1OEM-4>ID,EKONCT,W1MRA*,WI   W1OEM-4>ID,EKONCT,W1MRA*,WI
K1SEM>APWW10,KB1JDX-15,W1MR   K1SEM>APWW10,KB1JDX-15,W1MR   K1SEM>APWW10,KB1JDX-15,W1MR   K1SEM>APWW10,KB1JDX-15,W1MR
                                                            N1NCI-3>APN383:!4302.00NN07   N1NCI-3>APN383:!4302.00NN07
AB1OC-10>DX,RFONLY: <<UI>>:                                 AB1OC-10>DX,RFONLY:DX de  A   AB1OC-10>DX,RFONLY:DX de  A
N3LEE-7>T2TS4Y,WIDE1-1*,WID   N3LEE-7>T2TS4Y,WIDE1-1*,WID   N3LEE-7>T2TS4Y,WIDE1-1*,WID   N3LEE-7>T2TS4Y,WIDE1-1*,WID
N3LEE-7>T2TS4Y,WIDE1-1,AB1O   N3LEE-7>T2TS4Y,WIDE1-1,AB1O   N3LEE-7>T2TS4Y,WIDE1-1,AB1O   N3LEE-7>T2TS4Y,WIDE1-1,AB1O
KE5KTU-9>TR2P3Y,W1CLA-1,WID                                 KE5KTU-9>TR2P3Y,W1CLA-1,WID   KE5KTU-9>TR2P3Y,W1CLA-1,WID
KE5KTU-9>TR2P3Y,W1MHL,W1MRA                                 KE5KTU-9>TR2P3Y,W1MHL,W1MRA   KE5KTU-9>TR2P3Y,W1MHL,W1MRA
                                                                                          KE5KTU-9>TR2P3Y,W1MHL,W1MRA
N1LMA-8>APNX01,EKONCT,W1MRA   N1LMA-8>APNX01,EKONCT,W1MRA   N1LMA-8>APNX01,EKONCT,W1MRA   N1LMA-8>APNX01,EKONCT,W1MRA
N1YG-1>T1SY9P,W2DAN-15,W1MR                                 N1YG-1>T1SY9P,W2DAN-15,W1MR   N1YG-1>T1SY9P,W2DAN-15,W1MR
WM1X>APU25N,WIDE2-1: <<UI>>   WM1X>APU25N,WIDE2-1 <UI C>:   WM1X>APU25N,WIDE2-1:@310423   WM1X>APU25N,WIDE2-1:@310423
                                                            WM1X>APU25N,AB1OC-10,WIDE2*   WM1X>APU25N,AB1OC-10,WIDE2*
```

… The totals for each are displayed once every 30 minutes. …

```
K1SEM>APWW10,N1NCI-3,WIDE1,                                 K1SEM>APWW10,N1NCI-3,WIDE1,   K1SEM>APWW10,N1NCI-3,WIDE1,
N1OMJ>APWW10,W1MRA*,WIDE2-1   N1OMJ>APWW10,W1MRA*,WIDE2-1   N1OMJ>APWW10,W1MRA*,WIDE2-1   N1OMJ>APWW10,W1MRA*,WIDE2-1
N1OMJ>APWW10,W1MRA,N8VIM,WI   N1OMJ>APWW10,W1MRA,N8VIM,WI   N1OMJ>APWW10,W1MRA,N8VIM,WI   N1OMJ>APWW10,W1MRA,N8VIM,WI
                              N1OMJ>APWW10,W1MRA,W1JMC* <   N1OMJ>APWW10,W1MRA,W1JMC*:@   N1OMJ>APWW10,W1MRA,W1JMC*:@
N1OMJ>APWW10,W1MRA,AB1OC-10                                 N1OMJ>APWW10,W1MRA,AB1OC-10   N1OMJ>APWW10,W1MRA,AB1OC-10
N1ESA>TQRX9P,EKONCT,W1MRA*,   N1ESA>TQRX9P,EKONCT,W1MRA*,   N1ESA>TQRX9P,EKONCT,W1MRA*,   N1ESA>TQRX9P,EKONCT,W1MRA*,

Totals after 30 minutes, KPC3+ 208, D710A 197, DireWolf-0 297, DireWolf-1 318

                              MCU-4>T1RS0S,EKONCT,W1MRA,W   MCU-4>T1RS0S,EKONCT,W1MRA,W   MCU-4>T1RS0S,EKONCT,W1MRA,W
                                                            N3LEE-7>T2TS5Q,AB1OC-10,WID   N3LEE-7>T2TS5Q,AB1OC-10,WID
N3LEE-7>T2TS5Q,AB1OC-10,WID   N3LEE-7>T2TS5Q,AB1OC-10,WID   N3LEE-7>T2TS5Q,AB1OC-10,WID   N3LEE-7>T2TS5Q,AB1OC-10,WID
K1SEM>APWW10,KB1JDX-15,W1MR   K1SEM>APWW10,KB1JDX-15,W1MR   K1SEM>APWW10,KB1JDX-15,W1MR   K1SEM>APWW10,KB1JDX-15,W1MR
                                                            W1AST>TRPR4S,KB1AEV-15,N1NC   W1AST>TRPR4S,KB1AEV-15,N1NC
AB1OC-10>DX,RFONLY: <<UI>>:   AB1OC-10>DX,RFONLY <UI R>:D   AB1OC-10>DX,RFONLY:DX de  A   AB1OC-10>DX,RFONLY:DX de  A
W1MV-1>BEACON,W1MRA*,MA2-1:   W1MV-1>BEACON,W1MRA*,MA2-1    W1MV-1>BEACON,W1MRA*,MA2-1:   W1MV-1>BEACON,W1MRA*,MA2-1:
```

… After running several hours and we find these totals: …

```
AB1OC-10>APWW10,N8VIM,WIDE1   AB1OC-10>APWW10,N8VIM,WIDE1   AB1OC-10>APWW10,N8VIM,WIDE1   AB1OC-10>APWW10,N8VIM,WIDE1
W1BRI>APW261,W1MRA*,WIDE2-1   W1BRI>APW261,W1MRA*,WIDE2-1   W1BRI>APW261,W1MRA*,WIDE2-1   W1BRI>APW261,W1MRA*,WIDE2-1
N1LMA>APU25N,EKONCT,W1MRA*,                                 N1LMA>APU25N,EKONCT,W1MRA*,   N1LMA>APU25N,EKONCT,W1MRA*,
N1LCY>APU25N,W1MRA*,WIDE2-1   N1LCY>APU25N,W1MRA*,WIDE2-1   N1LCY>APU25N,W1MRA*,WIDE2-1   N1LCY>APU25N,W1MRA*,WIDE2-1
N1ESA>TQRX9Q,EKONCT,W1MRA*,   N1ESA>TQRX9Q,EKONCT,W1MRA*,   N1ESA>TQRX9Q,EKONCT,W1MRA*,   N1ESA>TQRX9Q,EKONCT,W1MRA*,
AB1OC-10>APWW10,WIDE1-1*,WI   AB1OC-10>APWW10,WIDE1-1*,WI   AB1OC-10>APWW10,WIDE1-1*,WI   AB1OC-10>APWW10,WIDE1-1*,WI
AB1OC-10>APWW10,WIDE1-1,N8V   AB1OC-10>APWW10,WIDE1-1,N8V   AB1OC-10>APWW10,WIDE1-1,N8V   AB1OC-10>APWW10,WIDE1-1,N8V
                                                                                          AB1OC-10>APWW10,WIDE1-1,AB1
N1OMJ>APWW10,W1MRA*,WIDE2-1   N1OMJ>APWW10,W1MRA*,WIDE2-1   N1OMJ>APWW10,W1MRA*,WIDE2-1   N1OMJ>APWW10,W1MRA*,WIDE2-1
N1OMJ>APWW10,WA1PLE-15,WIDE   N1OMJ>APWW10,WA1PLE-15,WIDE   N1OMJ>APWW10,WA1PLE-15,WIDE   N1OMJ>APWW10,WA1PLE-15,WIDE
N1ZZN>APAGW,W1MRA*,WIDE2: <   N1ZZN>APAGW,W1MRA*,WIDE2 <U   N1ZZN>APAGW,W1MRA*,WIDE2:@3   N1ZZN>APAGW,W1MRA*,WIDE2:@3
K1SEM>APWW10,N1NCI-3,WIDE1,   K1SEM>APWW10,N1NCI-3,WIDE1,   K1SEM>APWW10,N1NCI-3,WIDE1,   K1SEM>APWW10,N1NCI-3,WIDE1,
K1SEM>APWW10,N1NCI-3,WIDE1,   K1SEM>APWW10,N1NCI-3,WIDE1,   K1SEM>APWW10,N1NCI-3,WIDE1,   K1SEM>APWW10,N1NCI-3,WIDE1,
N3LEE-2>GPS: <UI>:!4243.50N   N3LEE-2>GPS <UI>:!4243.50NS   N3LEE-2>GPS:!4243.50NS07144   N3LEE-2>GPS:!4243.50NS07144
```

```
Totals after 540 minutes, KPC3+ 4255, D710A 4076, DireWolf-0 5667, DireWolf-1 6088
```

Results

If we give the highest number a score of 100% and scale the others proportionally, the scores are:

- 100%	Dire Wolf, single bit fix up
- 93%	Dire Wolf, error-free frames only
- 70%	Kantronics KPC-3 Plus
- 67%	Kenwood TM-D710A

The exact proportions will vary depending on what stations you happen to hear.

There should be enough detail here for others to repeat this experiment and verify results.