

# APRS from the Bottom Up

Alan Crosswell, N2Y GK

December 30, 2002

## Introduction

This article attempts to introduce Bob Bruninga's APRS with a bottom up approach, starting with the 1200 baud VHF FM packet and working up through the APRS protocol, messages, hardware and software implementations.

## VHF FM Packet

### Analog to Digital: AFSK

Audio frequency shift keying (AFSK) is a simple modulation scheme that was first used in 300 and 1200 baud telephone modems. A *mark* or 1 bit is represented by a frequency of 1200 Hz. A *space* or 0 bit is represented by a frequency of 2200 Hz (or perhaps vice-versa). The transition between each successive mark or space waveform happens at a rate of 1200 baud. The frequencies of 1200 and 2200 Hz fit within the standard narrowband FM audio passband used for voice, so that AFSK is accomplished by simply generating 1200 and 2200 Hz tones and feeding them in to the microphone input of a standard voice transmitter.

The physical layer encoding of binary data into analog waveforms enables one to send streams of bits. Those familiar with PSK31, RTTY and some of the other digital modes will recognize that these bit streams can be sent in groups of 8 bit bytes to just send text characters. However, packet applies more structure than this, which is the subject of the next higher layer – packaging the bits into recognizable *frames* or *packets*.

You can find a sample of what AFSK sounds like (along with samples of many other digital modes at <http://www.kb9ukd.com/digital/><sup>1</sup>.

---

<sup>1</sup><http://www.kb9ukd.com/digital/>



Figure 1: AFSK waveform



Figure 2: AX.25 Frame

## AX.25 Frames

The designers of amateur packet radio based their implementation on the dominant commercial packet network protocol of the time, ITU standard X.25. Some modifications had to be made to X.25 mostly to accommodate the use of amateur callsigns for packet addresses. This protocol was named AX.25 although there is no ITU standard for it. The standard instead is published by TAPR<sup>2</sup>.

AX.25 organizes groups of bits on the wire into frames (or packets). Each frame has a definite structure as shown in figure 2. The frame is logically broken up into the following fields:

**Flag** The flag is a delimiter between frames. The 01111110 pattern is unique due to bit-stuffing (any time five 1's are seen, a zero is stuffed and vice-versa for decoding). Extra Flags are permitted between frames. This gives receiver time to sync up to the received signal and also allows transmitter to run continuously if it has to.

**Address list** The address list is between 14 and 70 octets (2 and 10 callsigns) and consists of a destination, source, and up to 8 intermediate digipeaters. The address is 7 octets consisting of the callsign followed by a 4-bit substation ID (SSID) and 4 flag bits. Flag bits of note include the *repeated* and *end of list* (last digipeater) markers.

**Control** is used mostly for AX.25 connection-oriented protocol.

**PID** The protocol ID identifies what higher level protocol is the frame carries data for. Examples include:

- AX.25 layer 3 (virtual circuits – connections)
- Internet Protocol (IP frames inside UI frames)
- Address Resolution Protocol (callsign-to-IP address)
- No layer 3 (UI frames)

**Information** is the "text" of the message.

**FCS** is the checksum used to detect garbled packets so they can be ignored.

While the AX.25 frame contains binary data (including such hokey things as shifting the ASCII characters of the callsign left by one bit to make even that unreadable), it is commonly represented in a human-readable form as implemented by the TAPR TNC-2 MONitor command:

```
source>dest,digi1,digi2*,...,digi8:text
```

```
K2YNT>APRS,RELAY,WIDE*,WIDE:!4033.24N/07421.21W
```

<sup>2</sup><http://www.tapr.org/tapr/html/ax25.html>

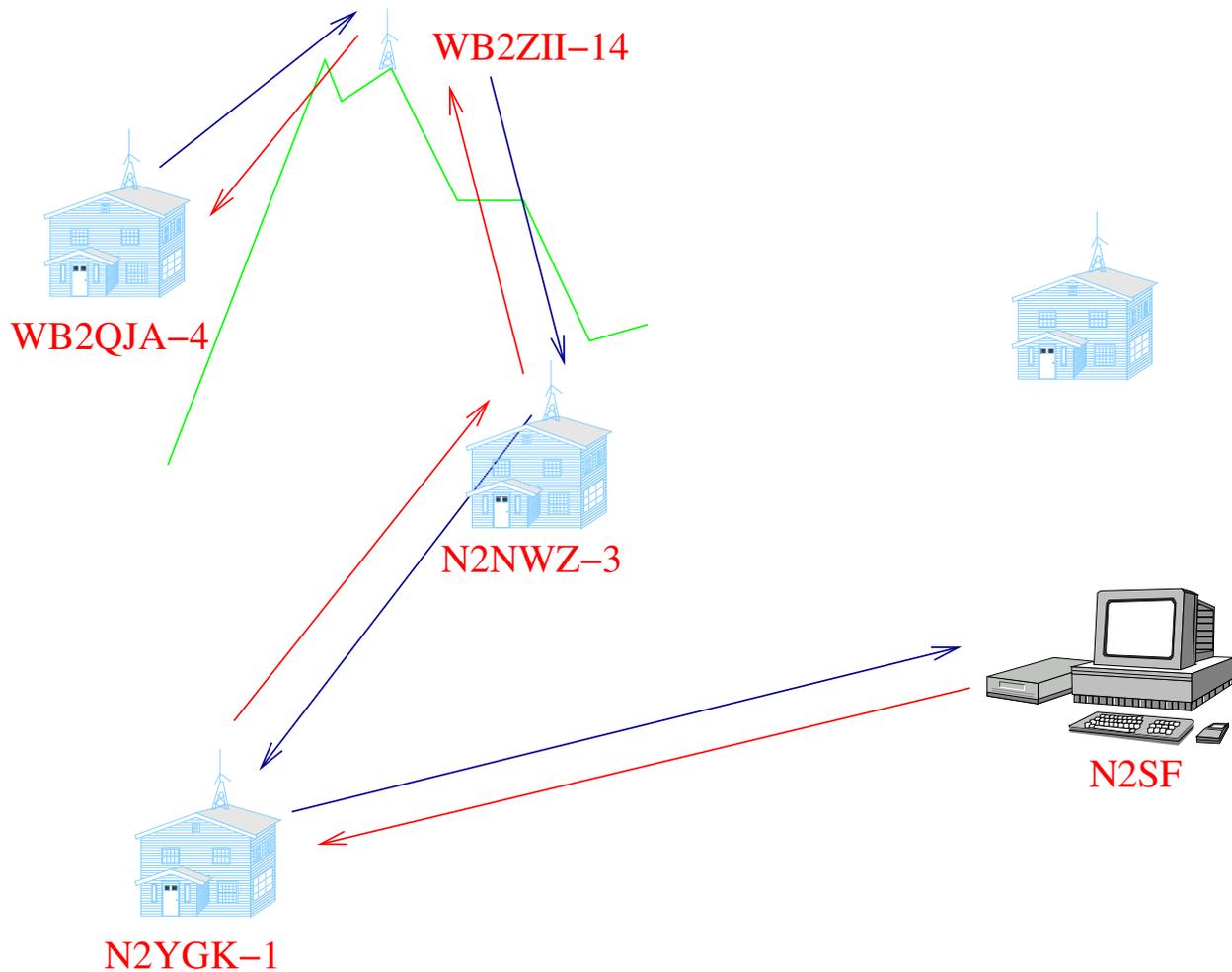


Figure 3: AX.25 Virtual Circuit

The “\*” indicates the last repeated digipeater. (All the preceding digipeater callsigns also have the Repeated flag set but it is not displayed.)

### AX.25 Connection Oriented

The conventional main use for AX.25, such as when connecting to a PBBS uses the Connection-Oriented (or virtual circuit) mode of AX.25. In this mode, the Control field and other specialty AX.25 frames are used to manage a virtual circuit connection between two stations, potentially with a number of intermediate digipeaters as illustrated in figure 3.

C WB2QJA-4 via N2YFK-1,N2NWZ-3,WB2ZII-14

### Compared to Internet Protocol

Those familiar with the Internet Protocol might find these comparisons to AX.25 helpful:

- AX.25 frames are like IP packets.
- AX.25 connection-oriented is like TCP.
- AX.25 connectionless (UI) is like UDP.

## APRS protocol's wierd use of AX.25

One of the key features of APRS, is that while it uses AX.25 to transport its messages, it essentially ignores all the AX.25 connection-oriented baggage.<sup>3</sup> All APRS messages are sent using connectionless Unnumbered Information (UI) frames. Furthermore, APRS has it's own data propogation model that uses a combination of broadcast flooding of messages, repetition, and special "well known" digipeater alias callsings.

### Broadcast flooding of messages

#### Special digipeater alias callsigns

Digipeater aliases are similar to tactical callsigns in a voice net (for example, "Logistics" or "Mile 11"). The digipeater alias is used to represent a specific function such as that of a PBBS or well-placed digipeater. For example, WECA has several nodes with callsigns WB2ZII-SSID that have aliases such as WECA5, WECA9, etc. The TAPR TNC-2 has always had this alias feature (MYAlias command).

Unlike typical packet use of aliases, in which a given single station has a specific alias, APRS uses some "well known" digipeater aliases that all or many stations use. The main two of these are *RELAY* and *WIDE*. It is not unusual for almost every APRS node to have *RELAY* as one of its aliases.<sup>4</sup>

With several stations all having the *RELAY* alias, a mobile APRS node can come into an area, transmit data with *RELAY* as the first digipeater alias to use, and be pretty likely to have that data digipeated by one or more nearby nodes. This is illustrated in figure 4. Luckily, digipeaters in the digi list are "eaten" up in order as they are used. Otherwise, with all nodes responding to *RELAY*, a packet could propogate forever. When a digipeater retransmits a packet, it marks the digipeater callsign that it responded to with the Repeated flag.

What about the converse: A mobile node transmits a packet and no *RELAY*s hear it? Perhaps the *RELAY*s are getting interference or are desensed by nearby transmitters? APRS addresses this problem by using repetition. The same data packet is sent several times, to make sure it gets there at least once. APRS also listens for a digipeat of its own packet to know that it made it out so it doesn't have to continue repeating it.

How widely one's APRS data is disseminated is controlled largely by how you set your digipeater path:

- Setting no digipeater path means only stations who can hear you directly will receive your data.

<sup>3</sup>This is very similar to how Amateur IP works – it also uses AX.25 minimally just to put IP packets on the air.

<sup>4</sup>Newer TNCs and software implementations can support more than just the single alias supported by the TNC-2.

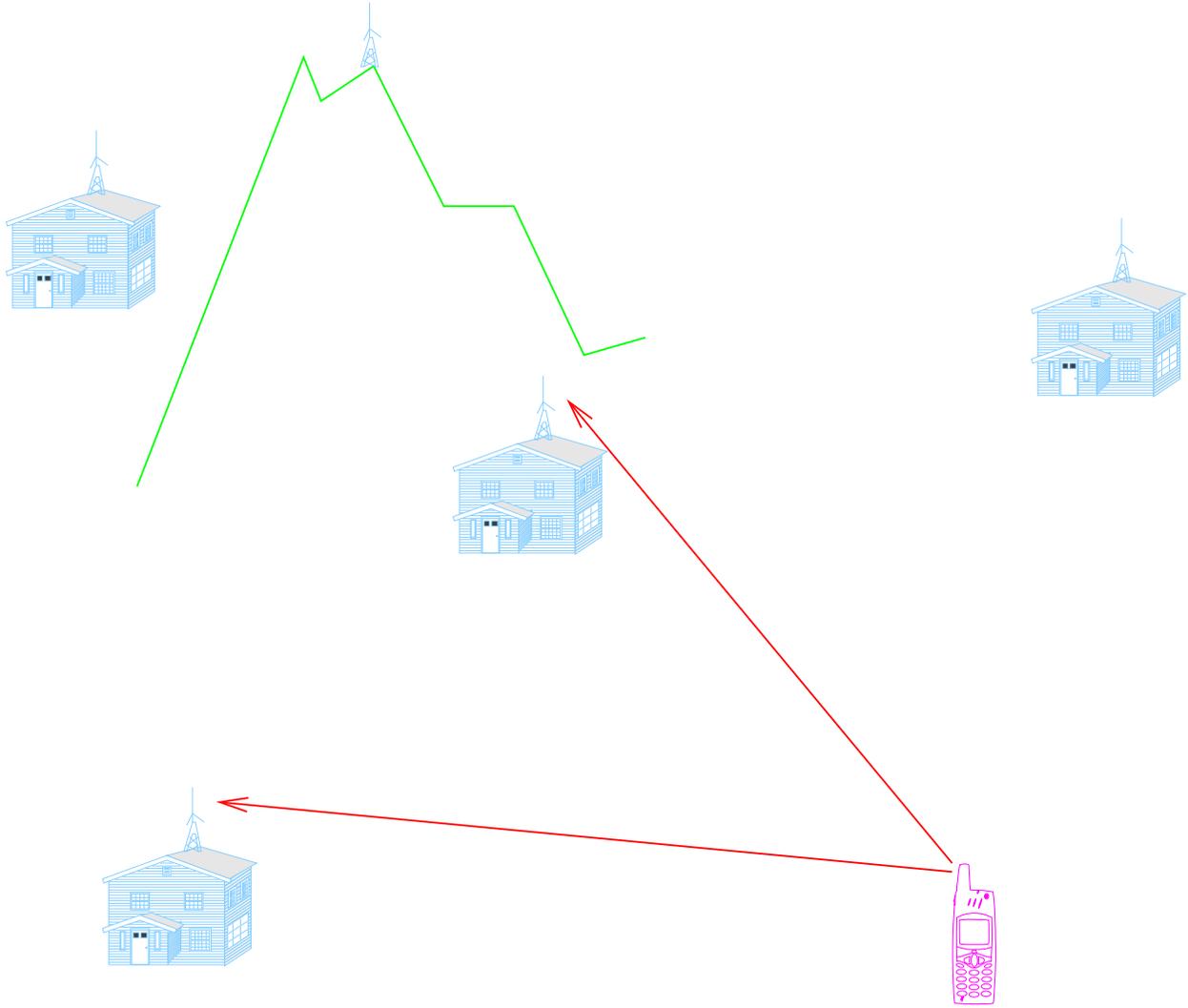


Figure 4: Broadcast Flooding: Time 1

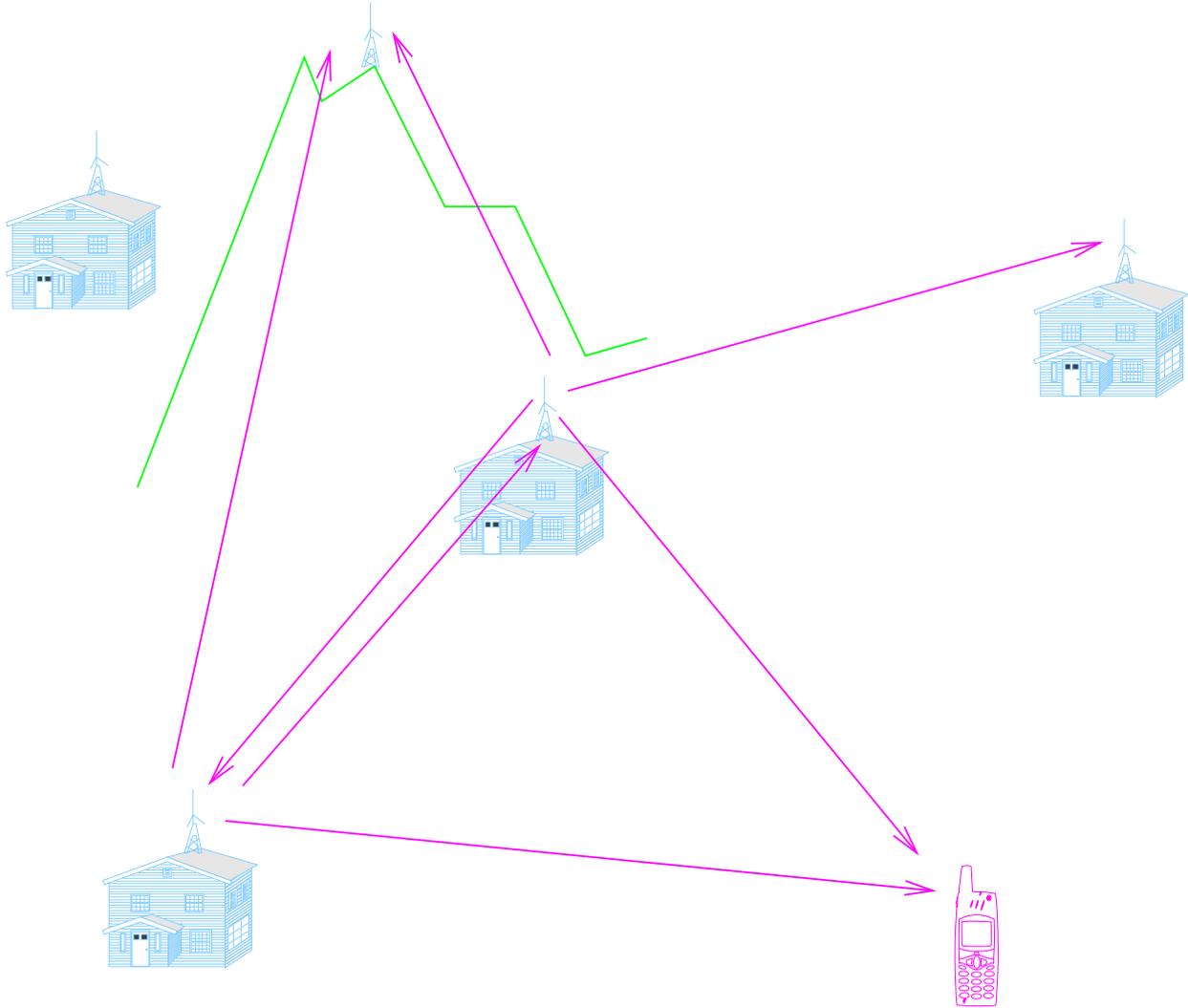


Figure 5: Broadcast Flooding: Time 2

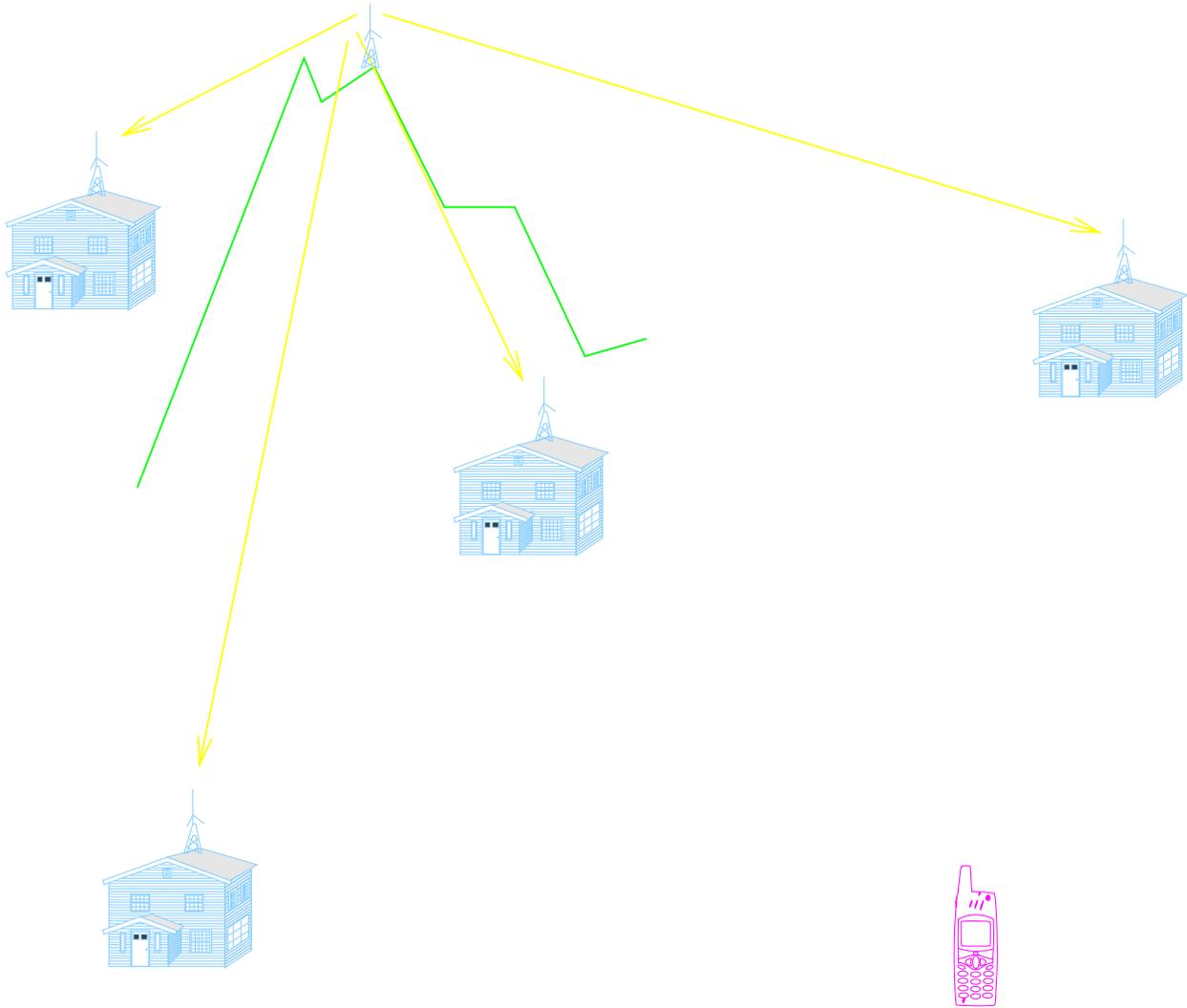


Figure 6: Broadcast Flooding: Time 3

- Setting a single hop path (typically RELAY) will mean the direct stations will in turn digipeat your data to all stations that can hear them directly.
- Setting a multiple hop path can expand your circle out even further. However, sending multiple hops all via the RELAY alias can lead to unnecessary repetition. Usually, after the first RELAY alias, one chooses an alias of WIDE. The idea is that wide coverage digipeaters are well spaced apart (that is, planned) unlike ad-hoc RELAYS.

### Alias callsign substitution

Many newer TNC's and software APRS digipeaters implement some more advanced features that address a problem caused by the use of generic aliases such as RELAY and WIDE: *Which* RELAY or WIDE was it? Since the alias callsign has been “eaten up” once its been used to digipeat a

packet, these newer digipeaters rewrite the alias back to the digipeater's real callsign. An example follows, in which N2YGK-9 and N2NWZ are both digipeaters listening for the RELAY alias and WB2ZII is a digipeater listening for WIDE:

```
K2YNT>APRS,RELAY,WIDE,WIDE:!4033.24N/07421.21W
```

```
K2YNT>APRS,N2YGK-9*,WIDE,WIDE:!4033.24N/07421.21W
```

```
K2YNT>APRS,N2NWZ*,WIDE,WIDE:!4033.24N/07421.21W
```

```
K2YNT>APRS,N2YGK-9,WB2ZII*,WIDE:!4033.24N/07421.21W
```

### **WIDEn-n aliases**

Another tweak made to digipeaters is the use of the WIDEn-n alias: Instead of a digipeater path, of, for example:

```
RELAY, WIDE, WIDE, WIDE
```

One makes a shorter digipeater path of:

```
RELAY, WIDE3-3
```

As each WIDEn-n digipeater is traversed, the SSID is counted down to zero. When n has counted down entirely to zero (WIDEn), the packet is repeated on final time, and the Repeated flag is set. This has the effect of setting a flooding radius and "remembering" what it was. For example:

```
K2YNT>APRS,N2YGK-9*,WIDE3-3:!4033.24N/07421.21W
```

```
K2YNT>APRS,N2YGK-9*,WIDE3-2:!4033.24N/07421.21W
```

```
K2YNT>APRS,N2YGK-9*,WIDE3-1:!4033.24N/07421.21W
```

```
K2YNT>APRS,N2YGK-9,WIDE3*:!4033.24N/07421.21W
```

You will note from the above example that there are two drawbacks to this approach:

1. Callsign substitution is not possible.
2. A much longer digipeater path could potentially be constructed (which is probably bad Amateur practice)!

## TRACEn-n aliases

The solution to the problem of getting a trace of WIDEn-n digipeats is to use TRACEn-n. In this case, each digipeater inserts its substituted callsign in the path – up to the maximum number of digipeaters allowed. If more digipeaters are needed, they simply replace the last one, so a long path that combines TRACEn-n and other aliases won't ever throw away any digipeaters that the originating station had put in the path. In the following example, WB2ZII, N2MH-15, and W2AEE are all TRACEn-n capable digipeaters:

```
K2YNT>APRS,N2YGK-9*,TRACE3-3:!4033.24N/07421.21W
```

```
K2YNT>APRS,N2YGK-9,WB2ZII*,TRACE3-2:!4033.24N/07
```

```
K2YNT>APRS,N2YGK-9,WB2ZII,N2MH-15*,TRACE3-1:!403
```

```
K2YNT>APRS,N2YGK-9,WB2ZII,N2MH-15,W2AEE*:!4033.2
```

## Routing

APRS digipeater paths can also be used to route from VHF to HF or via the Internet, for example. The variety of means of performing this routing is beyond the scope of this article.

## Duplicate packet suppression

Now that APRS has finessed flooding your data way too much, it also adds a means of suppressing too much flooding. Some APRS digipeaters keep a history of recently received packets (for the last 90 seconds, for example) and throw away any duplicates (based on the Information field remaining the same). Besides helping to solve a lot of looping flooding this technique also dampens the noise level coming from lids who set their APRS station beacon intervals to unreasonable short times (e.g. one a second)!

## APRS messages

We've now finished introducing the APRS digipeating infrastructure. What about the data that we've worked so hard to flood (and suppress)? APRS sends a variety of messages, including telemetry, short two-way text messages, bulletins, queries and replies.

### APRS messages: Telemetry

APRS sends out a variety of status messages which include the time, latitude, longitude, altitude, heading and speed. Other data can include transmitter power, height, and gain, DF bearings, weather conditions and a variety of other objects.<sup>5</sup>

---

<sup>5</sup>A well-known APRS example shows a chessboard off Bermuda.

Positional data can be manually entered or can be sourced by a GPS receiver. The encoding of the GPS data can either be direct NMEA<sup>6</sup> “sentences” or one of several possible APRS-specific encodings. Some examples follow:

```
KB2SCS>APRX46,WA2GUG-2,WIDE2*:>161726z kb2scs@arrl.net
```

```
N2YGK>APX104,RELAY,WIDE,WIDE: )Finish!4046.32N\07358.58Wm{Finish}
```

```
N2MH>GPSLJ,RELAY*,WIDE:$GPRMC,205532,A,4046.62,N,07416.12,W,000.0,050
```

```
N2YGK>APX104,RELAY,WIDE:=4108.97N/07349.14WxPHG5030/XASTIR-Linux
```

```
N2YGK>APX104,RELAY,WIDE:;Hot Patch_042114z4045.66N/07357.63Wk
```

Since APRS is trying to cram a lot of information into a fairly low bandwidth channel, a number of compression techniques have been developed. These include reusing the destination callsign (you may have noticed that it’s not really used for anything of importance) and compressing positional data to take up less space. An example of this is the APRS Mic-Encoder (Mic-E) compression shown in the following:

```
N2NWZ-4>T0TW4X,N2ZRC*,WIDE,WIDE: 'eU01" )v/ ]"4e}
```

The destination callsign in the above humanly-unreadable example contains an encoded version of the position, as does the remaining message text. Mic-E encoding also supports the concept of SSID-encoding the digipeater path (which must be supported by the digipeater receiving the packet). By SSID-encoding, the packet can be shortened further, completely eliminating the digipeat path resulting in, for example:

```
N2NWZ-4>T0TW4X-3: 'eU01" )v/ ]"4e}
```

As can be seen from table 1, the use of SSID 3 above is equivalent to using WIDE3-3 (which is roughly equivalent to a path of RELAY, WIDE, WIDE). Four bits of data already being sent results in 21 bytes of data not clogging the airwaves – saving 140 ms for each repetition of the packet. Also of note is that SSIDs 8-15 allows the APRS digipeater administrator to determine the best next-hop digis in the indicated directions. For example, WB2ZII might use N2MH-15 (in West Orange, NJ) as the next hop for the West path.

## **APRS messages: Two-way text**

So far, the APRS messages shown have been one-way announcements of a station’s location, etc. APRS also supports two-way reliable messaging. One can send a short text message to a specific station and that station, upon receipt of the message, will send back an acknowledgement:

---

<sup>6</sup>National Maritime Electronics Association

SSID	unproto path
0	none
1	WIDE1-1
2	WIDE2-2
3	WIDE3-3
4	WIDE4-4
5	WIDE5-5
6	WIDE6-6
7	WIDE7-7
8	NORTH UNPROTO path
9	SOUTH UNPROTO path
10	EAST UNPROTO path
11	WEST UNPROTO path
12	NORTH UNPROTO path + WIDE
13	SOUTH UNPROTO path + WIDE
14	EAST UNPROTO path + WIDE
15	WEST UNPROTO path + WIDE

Table 1: Mic-Encoder SSID-based digi path selection

```
N2Y GK>APX104,RELAY,N2MH-15*,WIDE::N2NWZ-2 :I'm tracking the trail now!{4
N2NWZ-2>APW246,WA2JNF-15,N2MH-15*::N2Y GK :ack4
```

If no ACK is received, the sending station periodically retransmits the message. One problem for APRS messaging is the appropriate selection of the correct digipeater path to get the message there. “Smart” implementations of APRS use the last received packet from the recipient to derive a reasonably good path.

## APRS messages: Bulletins

APRS Bulletins are one-way short messages. Rather than being addressed to a specific station, they are sent to the special callsign<sup>7</sup> BLN*n*. BLN1 is shown on line 1 of the bulletins display, BLN2 on line 2 and so on. Bulletins should be used sparingly or not at all:

```
KC2GMM>APR851,RELAY,WIDE,TRACE3-3::BLN1 :Welcome HAMS NYC marathon.{11}
KC2GMM>APR851,RELAY,WIDE,TRACE3-3::BLN1 :Anyone know freqs for race?{16}
```

## APRS messages: Queries/Replies

APRS supports a variety of general and directed (addressed to a single recipient) queries and their replies. Some of these include:

<sup>7</sup>Note that the destination callsign for messages and bulletins is not the AX.25 destination but rather embedded in the Information field. Go figure.

?**APRS** Query for what other APRS stations are on frequency. Is typically used when a new stations comes up and wants to get an up to date status. The query can be constrained to a circle around a given latitude and longitude in which case only stations within that circle reply.

?**APRSD** Asks for a list of stations heard direct. Useful for mapping out propagation given that the reporting stations typically have announced their location, altitude, and approximate EIRP.

?**APRSH** Asks if youve heard a particular station.

?**WX** Solicits weather telemetry from APRS stations equipped with automated weather measurement equipment.

## **APRS hardware**

### **“Classic” packet**

APRS runs on a large variety of hardware. It was originally developed as a BASIC program on a DOS PC using a TAPR TNC-2<sup>8</sup> connected to a standard VHF rig via the usual TNC to microphone in and speaker out connectors. Support was added for add-ons such as a serial-port GPS and a variety of weather stations. A simple hardware serial port sharing device was also designed to enable a PC such as a laptop with only a single serial port to share it between the TNC and GPS or WX station.

Of course, a TNC and transceiver are frequently set up as a standalone digipeater with no computer attached. This is how many of the original WIDE digipeaters are still configured today.

### **APRS-enhanced TNCs**

Once APRS took off as the “next big thing” in packet, TNC manufacturers added GPS support, enabling one to plug the GPS or a WX station into the TNC and have the TNC send that data to the APRS PC application via the TNC’s serial port, or to have the TNC operate as a standalone digipeater that also beacons GPS or WX data. Some TNCs were also manufactured with an add-in GPS receiver daughter board. Further firmware changes were made to some TNCs to support callsign substitution.

### **Non-TNC Transceiver add-ons**

The next round of APRS hardware innovation consisted of the elimination of the fairly cumbersome and expensive TNC,<sup>9</sup> replacing it with a small, portable, low-power transceiver add-on designed for mobile or portable use such as in vehicle “trackers” and balloons. These add-ons are universally a transmit-only solution and include three projects based on various models of the low-cost PIC microprocessor. These include the MIM, Mic-Encoder (MIC-E) and PIC-Encoder (PicE)

---

<sup>8</sup>Most commercial TNCs such as those made by Paccomm and MFJ are clones of the TNC-2.

<sup>9</sup>The TNC-2 is 1980’s vintage technology and consumes large amounts of power and uses some obsolete components, not to mention being physically large.



Figure 7: APRS MIC-E (beta)

which are all built around the PIC<sup>10</sup> microprocessor. PICs are very popular with hobbyists as they include all the basic I/O circuitry include serial and analog I/O, writable memory and RAM in a single chip.

The MIM, designed by Dr. Carl Wick, N3MIM, was a standalone tracker frequently connected to a transmit-only rig and initially designed for tracking balloons. The Mic Encoder (MIC-E) uses the same PIC processor as the MIM, has modified microcode, and was nicely packaged into a kit by TAPR<sup>11</sup>.

A key feature of the MIC-E is that it is designed to be inserted in the audio/PTT path of a standard mobile rig's microphone. It passes the voice and PTT signals through, but, upon release of the microphone's PTT, can insert a short APRS packet (while holding the rig's PTT signal). The result is that annoying "braaapp" you hear on the end of many of my transmissions on 06! That packet contains my vehicle's battery voltage, current GPS position, heading and speed, the value of an 8-position switch corresponding to standard message values of "Off duty", "Enroute", "In service", "Returning", "Committed", "Special", "Priority" and "Emergency" as well as my callsign and various special codes to select a digipeater path.

---

<sup>10</sup><http://www.microchip.com/>

<sup>11</sup><http://www.tapr.org>



Figure 8: APRS PIC-E

Some MIM and MIC-E users realized they could potentially be violating FCC rules by transmitting without listening, potentially interfering with other stations. Another feature of the MIC-E (and of course built as a add-on circuit to the MIM) is a “hold off” input, which can be wired to the squelch circuit. This is easily done with a Kenwood TM-733A mobile, for example, since the squelch signal is available on one of the pins of the data connector (that is typically used for connecting to a standard TNC-2).

One of the concerns raised by many in the APRS community was the closed source commercial nature of the MIM and MIC-E PIC software (as well as that of the TNC-2, produced twenty years earlier).<sup>12</sup> Customization of the hardware was easily done, but customization of the software, where all the work is happening, could not be done unless you happened to be Bob Bruninga and had the ear of the PIC software author. This led to the next TAPR kit, the PIC-E, which is an open source hardware and software implementation. The source code is published and Microchip makes their compilers and assemblers available free to anyone who buys their chips, so anyone with a mind to do so, or just wanting to learn how the smoke inside the chip works, could do so.

The first PIC-E code was written by Byon Garrabrant, N6BG and since then a number of authors have used his code as the basis for a variety of mods to the PIC-E (which is like a MIC-E) including support for a variable beaconing rate that is tied to the speed of the vehicle; two-way operation, including as an in-band or cross-band digipeater; Peet Ultimeater 2000 and Radio Shack WX-200; and adding on a 2K serial EPROM. See <ftp://ftp.tapr.org/picsig/software><sup>13</sup> for some of

<sup>12</sup>Not to mention all the APRS PC software with the exception of Xastir!

<sup>13</sup><ftp://ftp.tapr.org/picsig/software>

these examples and The TAPR kits page<sup>14</sup> for more info on the various kits available.

## APRS Rigs

If you prefer your APRS hardware nicely packaged and fully portable, there's the Kenwood [TH-D7](#) and the [TM-D700](#) mobile. Both of these rigs include a built-in 1200/9600 baud TNC-2 clone as well as full-blown built-in APRS functionality, including two-way operation. You can even interactively message. Connect a GPS to the TH-D7 or TM-D700's input jack and you have a complete APRS station. Or connect the rig to your computer and use one of the software APRS implementations.<sup>15</sup> For in person reviews you can talk to a number of WECA members who have the D7 and D700, and one, who even sprang for the slowscan TV accessory (that would be N2NWZ).

Alinco also makes the DR-135TP<sup>16</sup> which is a mobile rig with built-in TNC with basic GPS beaconing capability. This one could be ideal if you plan to run APRS PC software and just want a fully packaged radio/TNC. I recommend reading some reviews of this radio before running out and getting one since I don't have direct experience with it.

## APRS application programs

So far, we've concentrated on the underlying APRS tool set: the protocol and some hardware that runs it. There are a number of APRS programs available that display geographic information, support messaging, DFing, and so on. These programs run on the gamut of computer hardware from 286 PCs to Pentium 4's, Macs, and Palm Pilots. Which one to use is up to what hardware you have available, what OS it is running, and what features you want. Many of the APRS software versions are distributed as "nagware" that lets you run the software pretty much completely – until you want to save your configuration that you just painstakingly spent the last hour getting right. I encourage you to grab a copy of the various programs and try them before sending in the registration fee. Or, even better, use the free one on the free OS: Xastir on Linux. You can find most APRS downloads on the TAPR web site at <http://www.tapr.org><sup>17</sup>. A brief summary of the APRS versions I am aware of include:

**APRSDos** The original, by Bob Bruninga, WB4APR. Will run on any number of older DOS PCs.

**WinAPRS/MacAPRS** by Mark Sproul, KB2ICI and Keith Sproul, WU2Z, for Windows and Macintosh.

**APRS Plus** by Brent Hildebrand, KH2Z. Works with Delorme Street Atlas USA for Windows (older versions).

**palmAPRS** by Mike Musick, N0QBF. For the Palm pilot.

---

<sup>14</sup><http://www.tapr.org/tapr/html/kitsf.html>

<sup>15</sup>But don't be surprised if KISS mode doesn't work!

<sup>16</sup><http://www.alinco.com/Products/DR-135/>

<sup>17</sup><http://www.tapr.org>

**Xastir** Originally by Frank Giannandrea, KC2GJS, now with several authors. This program runs on many flavors of Unix including Linux, Solaris, and MacOS 10. Supports a gazillion map formats, speech output, etc., building upon a large library of free Unix software. See <http://www.xastir.org><sup>18</sup> and <http://xastir.sourceforge.net><sup>19</sup>.

## Advanced APRS apps

Beside the more typical PC desktop/laptop/palmtop APRS applications, there are a number of others, many of which integrate the Internet with APRS. Some examples of these applications include:

**javAPRS** By Steve Dimse, K4HG. A Java web viewer of Internet-connected APRS data collectors. Unfortunately Steve is also of the commercial software camp, so his code is unavailable to learn from or modify. **javAPRS** is now maintained by Pete Loveall, AE5PL at <http://www.aprs-is.net/javAPRS/><sup>20</sup>. See also, Steve's [findu.com](http://www.findu.com)<sup>21</sup> which you can use to search for APRS stations that have been "gated" to the Internet data collectors. For example, try this URL: <http://www.findu.com/cgi-bin/find.cgi?call=wb2zii><sup>22</sup>

**aprsd** [APRS data collector/Internet gateway](#) by Dale Heatherington, WA4DSY.

**aprsdigi** Linux "does everything" digipeater by me:-). Currently running at W2AEE. See My APRS page<sup>23</sup>

**tig2aprs** Generates APRSdos-format maps from US Census Tiger/line files. (This has been obsolete by better Xastir map support.)

## The WB2ZII 2 Meter Repeater/APRS gateway

Now that all those MIC-E, TH-D7, and TM-D700 users are happily braaping away on 147.06, what use is that unless your APRS display program is also listening on 06, rather than on the national APRS frequency of 144.39.

The MIC-E digipeater listens to the output of 147.660's receiver voter (which is the "best" audio chosen from the multiple remote receivers that is then input to the transmitter.) Any packets mixed in with the speech that are decoded are then retransmitted on 144.390. And, normal 144.39 digipeating is also done. This is accomplished on a shoestring budget currently with a pair of TNC-2s wired back to back. The receiver TNC-2 is running with a KISS EPROM which outputs all decoded packets in KISS<sup>24</sup> format to the serial port.

The digipeater TNC is running a TheNet X1J4<sup>25</sup> EPROM because this software is capable of

---

<sup>18</sup><http://www.xastir.org>

<sup>19</sup><http://xastir.sourceforge.net>

<sup>20</sup><http://www.aprs-is.net/javAPRS/>

<sup>21</sup><http://www.findu.com>

<sup>22</sup><http://www.findu.com/cgi-bin/find.cgi?call=wb2zii>

<sup>23</sup><http://www.users.cloud9.net/~alan/ham/aprs/>

<sup>24</sup><http://people.qualcomm.com/karn/papers/kiss.html>

<sup>25</sup><http://www.nordlink.org/>

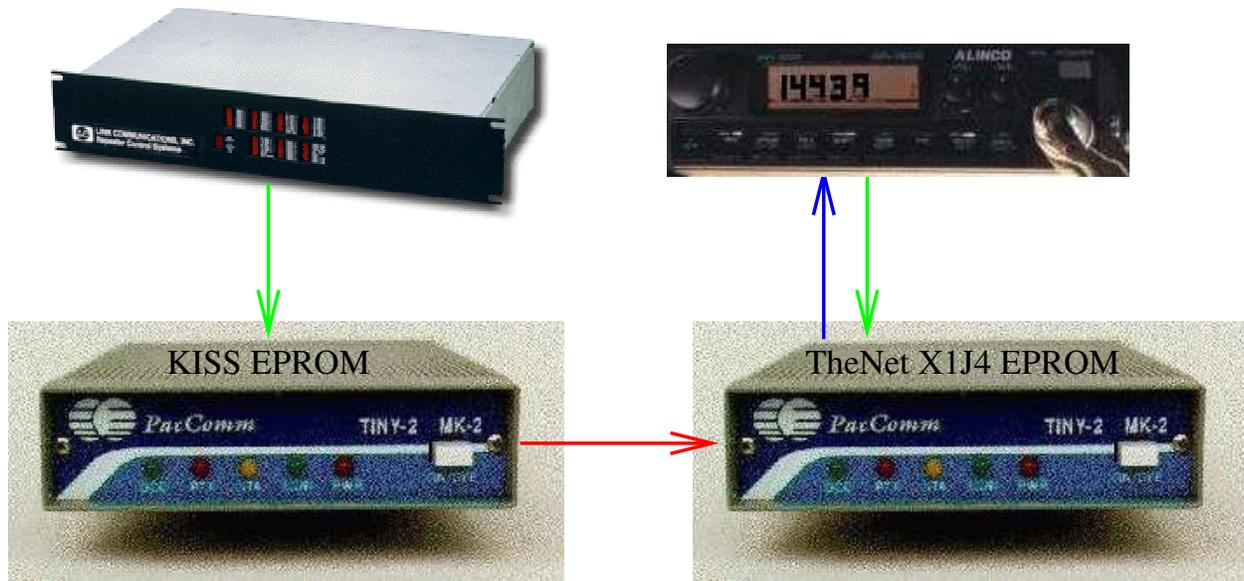


Figure 9: The WB2ZII Mic-E Digipeater Diagram

being set up to do standard UI digipeating and to transmit any KISS-formatted packets received on its serial port.

## Some Further Tips

During the presentation to the WECA membership several useful questions and answers were offered by the folks attending. These are summarized here:

### What's a good GPS to buy to use with APRS?

Check out any of the Garmin units. Even the extremely low-cost Etrex can be used, if all you want is a basic GPS without lots of fancy built-in or downloadable maps. Magellan units are also a good choice. Finally, if you don't want any display at all (e.g. for a permanent APRS-only installation) find a used Delorme Tripmate (they were only \$100 new). The replacement Earthmate only puts out Rockwell binary format (not NMEA) so you'll need to add a \$45 converter<sup>26</sup> at which point you might as well just buy a \$100 Etrex!

### Those darn Garmin serial cables are expensive. Is there an alternative?

Why yes, check out the Pfrancs<sup>27</sup> all over the world who trust you to send them money after they send you the inexpensive, high quality connector! Make sure you get the right one. Older Garmins had a round plug. The Etrex has a square plug. Both are available from your nearest Pfranc.

<sup>26</sup><http://www.byonics.com/gst-1/>

<sup>27</sup><http://pfranc.com/>



Figure 10: The WB2ZII Mic-E Digipeater at the Jail (and one of the UHF packet digis)

### Some more links

<http://www.weca.org/wcaprs/><sup>28</sup> has some info on APRS in Westchester County. See also my article<sup>29</sup> on APRS in the July, 1997 WECAGRAM.

---

<sup>28</sup><http://www.weca.org/wcaprs/>

<sup>29</sup><http://www.weca.org/wecagram/9707/wecagram.html>