APRS Digipeater Algorithm

John Langner, WB2OSZ
Rough Draft 1 - August 2022

Amateur radio repeaters retransmit signals from other stations to increase their range.

Analog voice repeaters listen on one frequency and simultaneously retransmit the same signal on a different frequency.

AX.25 digital repeaters (digipeaters) use a "store and forward" approach. A packet is received, examined, then possibly modified and retransmitted. Usually it is retransmitted on the same radio channel but it is also possible for a multi-port digipeater to link multiple radio channels. Packets received on one channel can be retransmitted on different channels.

The ***APRS Protocol Reference 1.0.1*** doesn't describe how a digipeater should work. All we have is precedent from legacy TNCs, long before the WIDEn-N paradigm was created. As a result, we find many implementations that don't behave exactly the same. Newer implementations have found different ways to overcome weaknesses in the legacy TNCs.

1 The Standard "TNC-2" Monitoring Format

First we need to understand what the standard display format is telling us so we can understand the examples and actual packets seen over the air.

source > destination : information
source > destination , digipeater1 : information
source > destination , digipeater1, … , digipeater8 : information

The standard display format begins with 2 or more addresses:

Source address - Originating station.
Most often a ham radio callsign but it could be a tactical callsign. It can optionally be followed by a "-" and a number, maximum 15, called the substation identifier (SSID), to allow up to 16 stations to be operated with the same callsign.

Destination address - In traditional connected mode packet, this would be a specific station. In APRS, this is used in several different ways. We can ignore it for this discussion.

Via path - Up to 8 addresses for path that the packet has taken already and where it might go.

When a digipeater address is followed by "*", that one, and all earlier digipeater addresses, have been used up. They show the path of where the packet has been retransmitted already.

(Behind the scenes, in the AX.25 frame, each digipeater address has a "has been used" H bit to indicate that the address has already been used.)

2 Manual Routing

Suppose I wanted to explicitly route a packet through N2GH digipeater and then W2UB digipeater. The original packet would look like this, with 2 specific digipeaters listed. Notice that there is no "*" so we are hearing the original (source) station:

```
WB2OSZ>APRS,N2GH,W2UB:something
```

N2GH recognizes its name, in the **first unused** digipeater position, and retransmits the packet. The result would look like this:

```
WB2OSZ>APRS,N2GH*,W2UB:something
```

The N2GH digipeater sets the "has been used" flag (the AX.25 "H" bit), on the address to indicate that it has been used up and won't be considered for any future digipeating decisions. When you see "*" after a digipeater name, you know that you are hearing the transmission from there.

The same thing happens again. W2UB sees its name in the **first unused** digipeater position and retransmits the packet. You see the "*" after the callsign so you know that you are hearing that station.

```
WB2OSZ>APRS,N2GH,W2UB*:something
```

All of the digipeater addresses have been used up and this packet can't be retransmitted again.

Some software might display it like this with two "*" characters.

```
WB2OSZ>APRS,N2GH*,W2UB*:something          ← Wrong
```

It might seem more intuitive (both addresses have been used) but **it is wrong**. It does not conform to the rules of the standard monitoring format, where only the **last used digipeater** is marked with "*" and it is implied that earlier addresses have been used up. There should never be a used address after an unused address.

This is what you know if everyone is well behaved:

- The packet originally came from `WB2OSZ`.
- It was retransmitted by `N2GH`. (Therefore `N2GH` can hear `WB2OSZ`.)
- It was retransmitted by `W2UB`. (Therefore `W2UB` can hear `N2GH`.)
- You are hearing the transmission from `W2UB`. (It is followed by "*".)

From the AX.25 protocol specification:

> *"The destination station can determine the route the frame took to reach it by examining the address field and use this path to return frames."*

The second part might be true in theory but not always in practice. You could have a case where station X can hear station Y but Y can't hear X so the same reverse path won't work.

As we will see later, some implementations are not well behaved so we really don't know where the packet travelled. Either they don't identify themselves or leave useless junk in the path which creates an ambiguous situation.

3 What Gets Repeated?

Clearly a digipeater should not retransmit everything it hears. If it did that, anything that was heard on the channel would keep bouncing back and forth between all of the available digipeaters. First the sender needs to construct a suitable via path. The digipeaters need to have a suitable configuration specifying how they should behave.

Using specific station names is usually not very satisfactory. What digipeaters are available nearby? Who can hear me? Who can hear the digipeaters that hear me? What happens if my favorite digipeater is not available? What if I'm traveling and don't know what is in the vicinity?

"Aliases" can allow digipeaters to respond to additional names besides their own callsign. Multiple stations can respond to the same alias. For example, the local Emergency Operations Center (EOC) might respond to the alias "EOC" so you don't have to remember the exact callsign used. A digipeater on the top of Mount Washington might respond to the alias "MTWASH". Sometimes mobile digipeaters will respond to the alias "TEST."

The 20th Century legacy TNCs did not allow much flexibility, allowing at most 4 aliases. For example,

```
UIDIGI ON EOC,TEST
```

If I was to transmit something like this,

```
WB2OSZ>APRS,EOC:something
```

It might be retransmitted as:

```
WB2OSZ>APRS,KB1MKZ*:something        ← correct
```

The alias is always **replaced** by the callsign of the digipeater. It should never be retransmitted like this, with the alias marked as being used:

```
WB2OSZ>APRS,EOC*:something          ← wrong
```

Aliases are always replaced by the callsign of the digipeater. This gets back to the rule mentioned earlier that the used addresses should show you the path that the packet has taken, on its way from the source station, to you.

Two different old TNC manuals mentioned nothing about duplicate suppression for UIDIGI, as they do for UITRACE, so they might clutter up the radio channel with unnecessary duplicates of the same thing. Duplicate suppression will be discussed later.

4 The New n-N Paradigm

Most of the time, we don't want to manually specify a sequence of specific digipeaters. Instead we normally want to specify any digipeater or any with some property such as geographic region.

In the early days of APRS, digipeater aliases of "RELAY" and "WIDE" were used. This has been obsolete, since around 2004, and all uses of them should have been removed long ago. So let's not talk about them any more since it will only cause confusion.

Fixing the 144.39 APRS Network
The New n-N Paradigm
http://www.aprs.org/fix14439.html

The currently accepted method is to specify classes of APRS digipeaters in the generic form *XXXn-N*.

XXX	The prefix, up to 5 characters. Usually this is "WIDE" but others are allowed for geographical regions, special events, or other uses. For example, "MA" might be used for Massachusetts. "TEST" is sometimes enabled for mobile digipeater stations.
n	Usually 1 for a local "fill-in" short range digipeater. 2 is normally for a good location with long range. Theoretically numbers up to 7 can be used, but in practice only 1 and 2 are normally used.
N	The remaining number is maximum number of times the frame should be digipeated, also known as the hop count. Initially it is in the range of 1 thru 7. This is decremented until it reaches 0 and there is no more digipeating.

The 20th Century TNC doesn't allow much flexibility here.

```
UITRACE  WIDE,30
```

This means it will respond to an address composed of

The characters “WIDE”.
A digit in the range of 1 through 7.
An SSID in range of 1 through 7.

This is not very customizable. It will match 49 different combinations such WIDE1-1, WIDE1-7, WIDE2-2, WIDE7-5, etc.

There doesn’t seem to be a way to specify more than a single generic alias or to specify the number before the SSID.

The traditional digipeater configuration commands are inadequate for APRS after 2004. Newer implementations have come up with different approaches for more flexibility in configuring behavior.

5 Packets Gone Wild

If we are not careful, digipeating could get completely out of control. An original packet might get heard by several digipeaters and retransmitted by each of them. A larger growing ring of digipeaters hears multiple others and retransmits what it heard from each. The original station might hear its own packet resent by other stations and retransmit it, forming a loop.

There are a few things we can do to bring the situation under control.

5.1 Decreasing Hop Count

The originating station specifies the maximum number of times that a packet can be retransmitted. For example, we might start with:

WB2OSZ>XXXX,WIDE1-3:whatever

A digipeater configured to repeat for the pattern WIDE1-N would decrement the hop count, to 2, and insert its own call sign:

WB2OSZ>XXXX,WW1ABC*,WIDE1-2:whatever

Another digipeater would decrement the hop count to 1 and send:

WB2OSZ>XXXX,WW1ABC,WW2DEF*,WIDE1-1:whatever

Notice how there is only a single “*” after the last digipeater name. The next digipeater would decrement the hop count to send one of these two forms:

WB2OSZ>XXXX,WW1ABC,WW2DEF,W3GHI*:whatever

In the first example, the WIDE1-0 has been all used up so we can discard it and make the packet smaller. Some implementations leave the generic alias after it has been used up:

WB2OSZ>XXXX,WW1ABC,WW2DEF,W3GHI,WIDE1*:whatever

This creates an ambiguous situation. Did we hear W3GHI or did we hear some other station that did not identify itself? I think the first approach makes a lot more sense and follows the digipeater path intent in the AX.25 protocol specification:

> *"The destination station can determine the route the frame took to reach it by examining the address field and use this path to return frames."*

Anyhow, the digipeater addresses have been all used up so this packet can't be digipeated again.

5.2 Delay from clear channel to start of transmit

Usually, when we are preparing to transmit, we wait for a clear channel, and then wait a random amount of time to minimize the chances of transmitting at the same time as someone else. This is based on the SLOTTIME and PERSiST parameters.

For the usual default values, we have delays with the following probabilities:

Delay, mSec	**Probability**	
100	.25	= 25%
200	.75 * .25	= 19%
300	.75 * .75 * .25	= 14%
400	.75 * .75 * .75 * .25	= 11%
500	.75 * .75 * .75 * .75 * .25	= 8%
600	.75 * .75 * .75 * .75 * .75 * .25	= 6%
700	.75 * .75 * .75 * .75 * .75 * .75 * .25	= 4%
etc.	...	

If a signal is detected during this random wait time, we go back to the top and start over.

In the case of digipeating, we start transmitting **immediately** when the channel becomes clear. Rather than trying to avoid a collision, digipeaters immediately start transmitting at the same time on top of each other. The AX.25 protocol specification refers to these as "expedited" frames. Due to the FM capture effect, the strongest signal should win.

Legacy TNCs often have a parameter, called **UIDWAIT**, which needs to be **off** for this to work properly.

Using a KISS TNC for a digipeater is a bad idea because the KISS protocol has no way to apply "nowait" for individual packets. If you use a KISS TNC, and a separate application, for a digipeater, it will wait until after the well behaved digipeaters transmit at the same time, and possibly more poorly behaving digipeaters, and then cause extra unnecessary congestion.

5.3 Duplicate Suppression

The third part of the solution is to avoid sending duplicates within a certain amount of time, usually 30 seconds. A digipeater must remember everything it transmits and not transmit the same thing within 30 seconds. The comparison involves only the source, destination, and information part. In other words, the varying digipeater path is ignored when checking to see if two packets are the same.

A digipeater should not repeat a packet with its own callsign in the SOURCE field because that could result in a loop.

6 Digipeater Algorithm Summary

Digipeater configuration contains:

- Its own callsign.
- An optional set of aliases, additional names to which it will respond.
- An optional set of "generic" addresses where the SSID is a remaining hop count. Typically this will be WIDE1-N for a short range digipeater. It should be possible to had at least one more for geographical region or other special purpose.

6.1 Decide if it is eligible for repeating:

(a) Look for the first "unused" address in the digipeater addresses. Return NO if none found.

(b) Is the source my station address? If so, return NO.

(c) If the first unused digipeater address is my station address, return YES.

(d) If the first unused digipeater address is one of my aliases, return YES.

(e) If the first unused digipeater is of some generic form ***XXXXn-N*** matching a rule in my configuration, return YES.

The legacy UITRACE allows only a single prefix such as “WIDE” which would match “WIDE”, 1 thru 7, “-“, 1 thru 7.
Some later implementations offer greater flexibility including multiple prefixes and distinguishing “WIDE1” from “WIDE2”.

(f) Otherwise, return NO.

6.2 Suppress any duplicates.

(a) If we retransmitted this packet recently (typically within 30 seconds) then return NO.

The comparison is based on SOURCE, DESTINATION (excluding SSID), and INFORMATION parts. It does not include the digipeater addresses.
Implementations typically use a 16 or 32 bit hash, rather than keeping the entire string, to reduce storage space and compare time.

6.3 If it should be resent and the address is my call or one of my aliases:

(a) Replace the address with my address.
(b) Mark the address as used.
(c) Transmit it.

Or, if it should be resent and of the form XXXXn-N:

(a) If N >= 2, the N value is decremented. The digipeater callsign is inserted before it and marked used if we have less than the limit of 8 addresses.

Example: W9XYZ >APRS,WIDE2-2
Becomes: W9XYZ >APRS,WB2OSZ*,WIDE2-1

(b) If N = 1, we don't want to keep WIDEn-0 in the digipeater list so the generic address is replaced by the digipeater callsignand marked as used.

Example: W9XYZ >APRS,WIDE2-1
Becomes: W9XYZ >APRS,WB2OSZ*

(c) If N = 0, the hop count has been used up and the packet is not digipeated. This is an error condition that we should not encounter. If the address count was all used up, we would expect the has-been- used “H” bit to be set. There is at least one defective implementation out there which produces this.

6.4 Transmit and remember it

(a) Transmit the modified packet, WITHOUT the normal random delay after the channel is clear.

(b) Add to list of recently retransmitted packets along with a timestamp.