# APRS Digipeater Algorithm

John Langner, WB2OSZ
APRS Foundation
September 2024
Revised July 2025

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. You select the desired repeater with your transmit frequency and usually a CTCSS tone.

APRS 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.

How to select the desired digipeater(s) - or have it done automatically - will be discussed below.

The original APRS Protocol Reference 1.0.1 (2000) didn't say much about how a digipeater should work. Probably because everyone was still using 1980s style TNCs at the time. The current practice of the WIDEn-N paradigm was developed about 4 years after the original APRS standard was published. Someone wishing to implement a digipeater had only poorly documented inconsistent precedent from legacy TNCs, developed long before the WIDEn-N paradigm was created. As a result, we find many implementations that don't behave exactly the same. Some are just plain wrong.

Recommended background reading: [Understanding APRS Packets](#).

Digipeater behavior is now covered in the [APRS Protocol Reference 1.2](#) (2024) and described in more detail here.

## 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.<br>Most often a ham radio callsign but it could be a tactical callsign.<br>It can optionally be followed by "-" 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 because digipeaters don't look at it. |
| Digipeater **Via Path** | - Up to 8 addresses for path that the packet has taken already and where it might go.<br><br>**<u>When a digipeater address is followed by "*", that one, and all earlier digipeater addresses, have been used up.</u> They show the path of where the packet has been retransmitted already.**<br><br>*(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.)*<br><br>If this part is empty, the packet will not be retransmitted by any digipeaters. |

# 2 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.

APRS digipeaters look at the **first unused address** in the digipeater via path and retransmit the packet if the first unused address matches any of the following:

- The digipeater callsign.
- An alias (including SSID) for the digipeater callsign.
- An alias with the WIDEn-N paradigm.

## 2.1 Legacy AX.25 Manual Routing

This is a carryover from early days of amateur packet radio where you needed to be aware of the available digipeaters and explicitly specify digipeater(s) between your station and the desired destination.

As an example, suppose you wanted to 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>APZ,N2GH,W2UB:something
```

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

```
WB2OSZ>APZ,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>APZ,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 the digipeaters don't identify themselves or leave useless junk in the path which creates an ambiguous situation.

## 2.2 Aliases

"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 address alias <u>**includes any SSID**</u> so EOC, EOC-1, and EOC-2 could be used for different stations.

The 20th Century legacy TNCs typically allowed up to 4 aliases.  For example:

```
UIDIGI ON EOC-1,EOC-2,TEST
```

If I was to transmit something like this,

```
WB2OSZ>APZ,EOC-1:something
```

It might be retransmitted as:

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

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

```
WB2OSZ>APZ,EOC-1*: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.

## 2.3 The New n-N Paradigm Aliases

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?

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

In the early days of APRS, digipeater aliases of "RELAY", "WIDE", and "TRACE" were used. This has been obsolete, since around 2004, and all uses of them should have been removed long ago. I just mention this in case you see it used anywhere you will know it is wrong. So let's not talk about them any more since it will only cause confusion.

Fixing the 144.39 APRS Network<br>The New n-N Paradigm<br>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 routing prefix, up to 5 characters. Usually this is "WIDE" but others are allowed for geographical regions, special events, or other uses. For example, "MD" might be used for information of interest in Maryland but not other states.

*n* This is the digipeater role or class. Usually 1 for a local "fill-in" short range digipeater. 2 is normally for a good location with wide-area coverage. Numbers up to 7 can be used, but in practice only 1 and 2 are normally used.

You might see mention of this as being the initial maximum number of times the packet can be digipeated. That is wrong. If that was the definition, the common WIDE2-1 would not be valid.

*N* The remaining number of times the packet can be digipeated, also known as the remaining hop count. Initially it is in the range of 1 thru 7. This is decremented until it reaches 0 and it is all used up. Once this reaches 0, the address should be removed from the path rather than leaving clutter and creating ambiguity.

The Appalachian Trail Golden Packet event is an extreme case. The path "HOP7-7,HOP7-7" is used to send packets about 3,500 km with digipeaters on mountain tops. The point I'm trying to make is that digipeaters should be configurable, for special situations, and not hardcoded with just WIDE1 or WIDE2.

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

```
UITRACE  WIDE,30
```

This means it will respond to an address composed of

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

It will match 49 different combinations such WIDE1-1, WIDE1-7, WIDE2-2, WIDE7-5, etc.

This is customizable. There is no way to specify:
- more than a single generic alias (e.g. regional such as MD),
- to specify the number after the prefix (to distinguish fill-in from wide-area),
- or to limit the number of hops.

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

# 3 Packets Gone Wild

If we are not careful, digipeating could get completely out of control and completely clog up the radio channel. 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.

## 3.1 Decreasing Remaining Hop Count

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

    WB2OSZ>APZ,WIDE3-3:whatever

(Note: WIDE3 is not normally used in practice. It is used here to illustrate how the mechanism works.)

A digipeater configured to repeat for the pattern WIDE3-N would decrement the remaining hop count, to 2, and insert its own call sign:

    WB2OSZ APZ,WW1ABC*,WIDE3-2:whatever

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

    WB2OSZ APZ,WW1ABC,WW2DEF*,WIDE3-1:whatever

The next digipeater would decrement the hop count to send one of these two forms:

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

Notice how there is only a single "*" after the last digipeater name, not after each used address.

Some implementations leave the generic alias after the remaining hop count has gone to zero:

WB2OSZ>APZ,WW1ABC,WW2DEF,W3GHI,WIDE3*:whatever

The packet is unnecessarily longer, wasting air time. This creates an ambiguous situation. Did we hear W3GHI or did we hear some other station that did not identify itself? Discarding used up aliases makes 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.

## 3.2 Delay from clear channel to start of transmit

When we want to transmit, first we must wait for a clear channel. After that, we normally 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 typical 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.

Be suspicious of any implementation that does not generate a random delay after a minimum fixed delay.

But, there is an exception to this rule…

### 3.2.1 Delay when transmitting a digipeated packet

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.

> *"**Generally, all APRS digipeaters are supposed to transmit immediately and all at the same time.** They should NOT wait long enough for each one to QRM the channel with the same copy of each packet."*
>
> *"This reflects the design philosophy of APRS: digipeaters are expected to retransmit as soon as the channel is clear, without introducing random delays, to avoid unnecessary duplication and reduce channel congestion."*
>
> *"The goal is that a digipeated packet is cleared out of the local area in ONE packet time and not N packet times for every N digipeaters that heard the packet. This means no PERSIST times, no DWAIT times and no UIDWAIT times."*
>
> *"APRS wants to clear the channel quickly to maximize throughput."*
>
> \- - *WB4APR (SK)*

Legacy TNCs usually 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 "no random wait" 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.

A clever KISS TNC could eliminate the usual random wait if any of the digipeater addresses are marked as used.

## 3.3 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 (excluding SSID), and information part. In other words, the varying digipeater path is ignored when checking to see if two packets are the same.

### 3.4 Don't Digipeat Your Own Packets

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

## 4 Digipeater Algorithm Summary

Digipeater configuration contains:

- Its own callsign.
- An optional set of aliases (including SSID), to which it will respond.
- An optional set of generic addresses, with:
  - Routing prefix (e.g.  WIDE, TEST, MD, HOP)
  - Digipeater role (e.g. 1 for fill-in or 2 for wide-area)

Sone implementations may allow additional constraints or filtering based on the packet content

### 4.1 Decide if it is eligible for repeating:

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

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

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

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

(e) Otherwise, return NO.

### 4.2 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. Obviously, it does not include the digipeater addresses because they keep changing.

Implementations typically use a 16 or 32 bit hash value, rather than keeping the entire string, to reduce storage space and compare time.

(b) If your station address is in the SOURCE field then return NO.

(c) Otherwise return YES. (it should be digipeated if section 4.1 also returned YES)

## 4.3 If it should be retransmitted

### 4.3.1 Because it matched my station address or alias:

(a) Replace an alias with my address.
(b) Mark the address as used.

### 4.3.2 Or, if it of the form XXXn-N matching one of the configured XXXn routing prefixes:

(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 >APZ,WIDE2-2 |
| Becomes: | W9XYZ >APZ,WB2OSZ*,WIDE2-1 |

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

| | |
|---|---|
| Example: | W9XYZ >APZ,WIDE2-1 |
| Becomes: | W9XYZ >APZ,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 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. See "Understanding APRS Packets" referenced earlier.

## 4.4 Transmit and remember it

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

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