Understanding APRS Packets

John Langner, WB2OSZ - October 2024
Update August 2025

There are some outdated copies of this floating around.
Find the latest at https://github.com/wb2osz/aprsspec

Trying to understand APRS packets can be confusing. Some are fairly obvious but others are really obscure. This introduction will help you decipher these mysterious signals and spot errors. This deals only with the human readable TNC-2 monitoring format. It does not cover AX.25 in HDLC frame format.

My goal is to produce a more approachable introduction rather than telling people to read the protocol specification. This does not cover all of the possible features and is a guide for understanding rather than implementing. For that, you will need to refer to the original APRS protocol specification from 2000 and more than 20 years of addendum documents scattered about. Or you can find the original and the updates merged together here.

About 1/3 of this document is devoted to errors observed locally. Some are caused by flawed implementations; others are caused by user error. Read this document carefully if you don't want to end up as an example of what not to do. Please encourage APRS vendors to fix their defects to improve the user experience.

Contents

1 The Standard "TNC-2" Monitoring Format

First we need to understand what the standard display format is telling us. There is a variable length address part and an information part.

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

The address part has:

- Source - Origin of the packet.
- Destination - Usually identifies the type of system that generated the packet.
- Digipeater via path - List of digipeaters that a packet may traverse and/or has traversed already.

When sent over the radio, the AX.25 protocol requires that addresses must be:

- A combination of 1 through 6 upper case letters and digits.
- A number in the range of 0 through 15.

We will later see that this naming restriction is relaxed in other contexts.

1.1 Source Address

The Source address contains the name of the station that originally sent the packet. Usually, this will be a ham radio callsign but you will also see descriptive tactical callsigns. An optional number, called the Substation Identifier (SSID), allows up to 16 stations to be operated under the same callsign.

Valid Examples:

- N2GH
- N2GH-1
- N2GH-15

Invalid Examples:

- N2GH-0 When SSID is zero, it is not displayed.
- N2GH-16 Maximum is 15.
- N2GH -1 Spaces between the callsign and SSID.
- n2gh Must be upper case.

Six characters is a limitation of the AX.25 protocol designed in the early 1980s. There are now some countries that have 7 character callsigns. In this case, the full callsign could be put in the comment field for identification purposes.

1.2 Destination Address (usually device identifier)

The name is confusing. In connection-oriented packet radio it is the destination station for point to point communication.

APRS uses a connectionless one-to-many broadcast so this field is used for a different purpose. Most of the time this field will identify the type of system that generated the packet. This is sometimes called the "tocall" field. Calling it something like system type or device id would make more sense.

Common examples:

- APDW18
- APWW11
- APTT4

Often trailing digits will indicate a version. The traditional list of system types is at http://www.aprs.org/aprs11/tocalls.txt . This has not been updated since the end of 2021. The authority is now https://github.com/aprsorg/aprs-deviceid .

Exercise for the reader: Try looking up the examples above to find out what they mean.

Rules:

- System/device identifiers begin with "AP".
- Maximum of 6 upper case letters and digits.
- It must not be empty.
- SSID is not used.

Sometimes the literature might have "APRS" in this field. **That is not to be taken literally**; it is a placeholder where you should put the system type.

1.2.1 Destination Address SSID

I wanted this to quietly fade away. But then, there was some discussion about using this for LoRa APRS to make the initial packet shorter. So, this deserves some discussion.

One of the first APRS trackers https://web.tapr.org/product_docs/Mic-E/mic-edev/mic-e.manual.pdf had 4 DIP switches to set the destination address SSID. The idea was that digipeaters would translate a non-zero destination into a normal digipeater path as shown below:

- [SSID] AX.25 SSID, interpreted as:

0	Use conventional digipeater path, if any
1	WIDE-1 omnidirectional flooding
2	WIDE-2
3	WIDE-3
4	WIDE-4
5	WIDE-5
6	WIDE-6
7	WIDE-7

8	North path
9	South path
10	East path
11	West path
12	North path + WIDE
13	South path + WIDE
14	East path + WIDE
15	West path + WIDE

This was many years before the WIDEn-N form was invented. I always considered this to be obsolete and did not implement it in my digipeater. As an experiment, I handcrafted some APRS packets with 1 or 2 as the destination SSID. NO digipeaters, in my vicinity, responded to this. So, it looks like most, if not all, application developers have come to the same conclusion, that it is obsolete, and did not implement it.

1.3 Digipeater Via Path

VHF/UHF has a rather limited range if you are not in a good location. We use voice repeaters to extend the range of our voice communications. The repeater listens on one frequency and simultaneously transmits on another.

APRS uses a different type of repeater called a digital repeater ("digipeater"). Rather than using two frequencies the digipeater receives the entire packet, evaluates it, and **might** retransmit it on the same frequency.

You can specify the actual names of digipeaters, but most commonly, you will use aliases, of the form WIDEn-n, so you don't need to know anything about the network topology. Let's look at a typical example.

```
WB2OSZ>APDW18,WIDE1-1,WIDE2-1:information-part
```

1.3.1 Unused / Used Digipeater Addresses

When a packet is transmitted for the first time, it looks something like this:

WB2OSZ>APDW18,WIDE1-1,WIDE2-2:*information-part*

All of the digipeater addresses are "unused." There is no asterisk ("*") character in the digipeater address part so you know that you are hearing the original source station.

Suppose that digipeater N2GH retransmitted this packet. The result would be:

WB2OSZ>APDW18,**N2GH***,WIDE2-2:*information-part*

WIDE1-1 had a remaining hop count of 1 so it is discarded and replaced by the digipeater address. The asterisk ("*") after the digipeater address means that you are hearing that digipeater and the address has been used.

Suppose that digipeater W2UB retransmitted this packet. The result would be:

WB2OSZ>APDW18,N2GH,**W2UB***,WIDE2-1:*information-part*

W2UB is followed by * meaning that you are hearing that station and the address is now used. There should be at most one * in the digipeater field. It is implied that all earlier addresses are used. You might see some applications put * after each used address but this is wrong.

Noticed what happened here. WIDE2-2 had a remaining hop count of 2. The hop count is decreased by one so we end up with WIDE2-1 left over.

Suppose that digipeater WA2NRE retransmitted this packet. The result would be:

WB2OSZ>APDW18,N2GH,W2UB,**WA2NRE***:*information-part*

You are hearing WA2NRE. All of the digipeater addresses have been used up and this can't be digipeated anymore.

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.)
- It was retransmitted by WA2NRE. (Therefore WA2NRE can hear W2UB.)

From the AX.25 protocol spec:

> ***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 might not work.

1.3.2 Digipeater Algorithm

Digipeaters look at the first unused address in the digipeater via path of a packet.

If the first unused address is the same as the digipeater, the address is marked used and the packet is retransmitted.

If the first unused address an alias for the digipeater, the alias is replaced by the actual name, address is marked used and the packet is retransmitted.

If the first unused address matches the pattern *XXXn-N*, and the prefix *XXXn* is in the in the digipeater configuration, the packet might be retransmitted. The processing depends on the value of the SSID, represented as *N* here.

N = 0 Do not digipeat.

N =1 Digipeat and replace generic form with digipeater callsign. Digipeaters should always identify themselves, and mark their address used, so the actual path taken is known.

N ≥ 2 Decrement N and leave address in path. Leave it marked as unused. Insert digipeater callsign before it and mark digipeater callsign as used.

The list of stations, in the used part of the digi via path (before the "*" character in the human readable monitoring format), reveals the path the packet has taken.

Digipeaters must keep a copy of all packets that it transmitted during the past 30 seconds and not send a duplicate during this time. The duplicate test is based on everything other than the digi via path.

A digipeater should only change the digipeater via path. It should NEVER change the source, destination, or information part. Corrupting the packet would break duplicate suppression.

1.4 Information Part

Everything after the addresses is called the Information Part. The first character is the Data Type Indicator (DTI) which indicates the format of the data following. A few examples:

- ! = / @ Position Report

- ‘ ` MIC-E Position Report
- ; Object Report
- : “Message”
- T Telemetry Data
- } Third-party traffic

Notice that the only letter defined is “T” for Telemetry Data. All other letters and all digits are invalid.

In general, the following fields are fixed width printable ASCII characters. We will explore the more common types below.

Some of the packet types allow a variable length free form comment at the end. These are not restricted to ASCII. Multi-byte sequences can represent Unicode characters in UTF-8 format.

Carriage return (0x0d) or line feed (0x0a) should not appear at the end. The APRS protocol specification says nothing about their possible appearance and this could cause interoperability issues. IGate stations will remove any CR/LF characters from the end.

2 Packet Types

The packet type is generally (with many special cases) identified by the first character in the Information Part. Some of the most common Data Type Indicators (DTI) are:

- ! = / @ Position Report
- ‘ ` MIC-E Position Report
- ; Object Report
- : “Message”
- T Telemetry Data
- } Third-party traffic

2.1 Position Report

This is probably the most common type of packet you will see. A station is sending its own location and other information. There are 4 variations, based on the Data Type Indicator.

!	no timestamp,	no APRS messaging.
=	no timestamp,	capable of APRS messaging.
/	with timestamp,	no APRS messaging.
@	with timestamp,	cabable of APRS messaging.

Timestamps are very rarely used because the current time is implied. Timestamps can be added by the receiving station if the received packets are being logged.

The intention was for the type of Position Report to convey whether the station was capable of APRS messaging.

Example:

W1KU-2>APDW16,W1MRA,N3LLO-3*:!4220.00N/07138.00W-PHG2020Northborough MA

Let’s break this down into the individual components.

W1KU-2	The source station.
APDW16	The packet is generated by Dire Wolf version 1.6.
W1MRA,N3LLO-3*	The packet has been retransmitted by W1MRA then again by N3LLO-3. The “*” indicates the station that we are hearing.
!	Data Type Indicator (DTI) meaning Position Report, no timestamp, no messaging.

42	Latitude in degrees. Must be exactly 2 digits.
20.00	Latitude in minutes. Must be exactly 2 digits, decimal point, and 2 fractional digits.
N	Hemisphere for latitude. Must be upper case N or S.
/	Symbol Table or Overlay.
071	Longitude in degrees. Note the leading zero. Must be exactly 3 digits.
38.00	Longitude in minutes. Must be exactly 2 digits, decimal point, and 2 fractional digits.
W	Hemisphere for longitude. Must be upper case W or E.
-	Symbol.

The combination of Symbol Table/Overlay and the Symbol represents the type of station and an icon displayed on a map. "/-" represents a house.

You can get a complete list of symbols at http://www.aprs.org/symbols.html or by running direwolf with the "-S" command line option.

Anything after that is called the Comment. Certain character combinations can represent optional types of data. The rest is free-form text. We will cover comments, in more detail, in a later section.

2.2 MIC-E Position Report

The MIC-E format was an attempt to make the packet as short as possible. Example:

N1JCM-9>TRQP7T,WA1PLE-4*:`c'wl|+>/`"4-}_%<0x0d>

The fields are broken down as follows:

N1JCM-9	The Source station name.
TRQP7T	All other packet types have the product identifier here. The MIC-E format Has the latitude and several other bits packed in here.
WA1PLE-4*	Used digipeater address.
`	Both ' and ` indicate the MIC-E format.

c'w	Longitude.
l|+	Speed and course.
>/	Symbol code & Symbol Table. Car in this case.
`	Message capable device identified by 2 character suffix.
"4-}	Optional Altitude at the beginning of comment field.
	Optional Comment here.
_%	Yaesu FTM-400DR
<0x0d>	Unexpected Carriage Return displayed in hexadecimal. This possibility is not mentioned in the official protocol specification so I would consider it to be a defect that all receiving stations need to work-around.

MIC-E format made the unfortunate design decision to use some ASCII control characters that are not printable. Example:

N1YOQ-1>TRUW5X,UNCAN*,WIDE2-1:`c9r<0x1c><0x1f>;#/"5D}Solar Powered Digipeter

Some systems will display the unprintable characters in hexadecimal like this: <0x1c> <0x1f> . Others might display some strange non-ASCII characters or nothing at all.

2.2.1 MIC-E Device Identifier

In the previous section we saw that MIC-E format position reports begin with either ‘ or ` then 8 characters in fixed positions. After this is an optional comment.

According to the documentation, the original MIC-E put a space between the fixed part and the comment:

xxxxxxxx (space) *comment* Original MIC-E

Sometimes you will see this form, without anything between the fixed part and the comment. In this case, we don’t know what generated the packet.

xxxxxxxx (no space) *comment* Unknown

When Kenwood adopted the MIC-E format, they put a different character before the comment field to identify the equipment type.

xxxxxxxx **>** *comment*	TH-D7
xxxxxxxx **]** *comment*	TM-D700

When later models were introduced, an additional character was placed at the end of the comment.

xxxxxxxx **>** *comment* **=**	TH-D72
xxxxxxxx **>** *comment* **^**	TH-D74
xxxxxxxx **>** *comment* **&**	TH-D75
xxxxxxxx **]** *comment* **=**	TM-D710

This is known as the legacy format. For later implementations, a new format is used.

- A prefix of ` indicates a system capable of messaging.
- A prefix of ‘ indicates not capable of messaging, e.g. a tracker.

At the end we now have a two character suffix for the manufacturer and model. Examples:

xxxxxxxx **`** *comment* **_(**	Yaesu FT2D (messaging capable)
xxxxxxxx **`** *comment* **_0**	Yaesu FT3D (messaging capable)
xxxxxxxx **`** *comment* **_3**	Yaesu FT5D (messaging capable)
xxxxxxxx **‘** *comment* **|3**	Byonics TinyTrack 3 (no messaging)
xxxxxxxx **‘** *comment* **|4**	Byonics TinyTrack 4 (no messaging)

Any optional altitude is at the beginning of the comment, after any device identifier prefix and before any suffix. Example:

xxxxxxxx **`** **"4-}** *comment* **_(**

https://github.com/aprsorg/aprs-deviceid has machine readable files which list the MIC-E prefix and suffix values.

Rather than hardcoding these rules, application developers are encouraged to read a file at runtime so users can update the device database without waiting for a new software release.

2.3 Object Report

Object Reports are very similar to Position Reports, except you are sending information about something other than yourself. The Data Type Indicator is “;” (semicolon).

Example:

W1OEM-5>APWW11,EKONCT,WA1PLE-4*:;ELYME *190116z4122.06N/07212.98W#145.03
Packet Node ELYME!W98!

Broken into individual parts:

W1OEM-5	Source station.
APWW11	System type that generated the packet.
EKONCT,WA1PLE-4*	We are hearing WA1PLE-4 after packet was repeated by EKONCT.
;	Semicolon is Data Type Indicator.
ELYME	Object name. Any printable ASCII characters including embedded spaces. Object name is case sensitive. Extra spaces must be appended to make field exactly 9 characters.
*	Asterisk ("*") for a live object. Underscore ("_") to kill it.
190116z	Timestamp. 19th day of the month. 01:16 UTC. Here is something really strange. The lack of a specific time is represented as 111111z which is a perfectly valid time. What was the reasoning behind that? Why not make it blank for lack of a timestamp?
4122.06N/07212.98W#	Location and symbol, same as Position Report. Symbol is "DIGI (white center)"
145.03	Looks like a frequency but it is not in standard format so it will not be processed properly. Discussed later.
Packet Node ELYME	Free-form text comment.
!W98!	Adds more resolution to the location. DAO is discussed later. Latitude = 41° 22.06 + 0.0090 N Longitude = 072° 12.98 + 0.0080 W

2.4 Telemetry

Telemetry is broken down into two types:

- The actual data - just numbers.
- Metadata: A description of how to interpret the numbers.

2.4.1 Telemetry Data

Telemetry data is indicated by "T" at the beginning of the information part. After that:

- "# and a three digit sequence number.
- Up to five numeric analog telemetry values.
- If all five are specified, they can be followed by eight binary values.

The original protocol specification, allowed only values of 000 through 255 which was adequate only for 8-bit analog to digital converters. A couple decades later, it was officially relaxed to be 000 thru 999.

Example:

N1YOQ-1>APMI0A,UNCAN,WIDE1*,WIDE2-1:T#196,174,000,000,000,000,00000000

In practice, longer variable length numbers with decimal points are common. Example:

W1HS-11>APMI06,N1LIT-6,WIDE2*:}N3LLO-2>APRX29,TCPIP,W1HS-11*:T#300,38.8,0.0,176.0,55.0,0.0,00000000

All of the modern applications that I tested understood this more flexible format. This is a case where the community agreed on an obvious improvement without waiting for the official standard to catch up.

2.4.2 Telemetry Metadata

Just a number, without and context, has little meaning. Is it temperature, solar cell voltage, or river water depth? Information describing data is called metadata.

The names/labels, units, and scaling coefficients are sent in a specific format of APRS "message" addressed to the station sending the telemetry data.

N1YOQ-1>APMI0A,N3LLO-3,WIDE1*,WIDE2-1::N1YOQ-1
:UNIT.Volt,None,None,None,None,On,On,On,On,Hi,Hi,Hi,Hi

N1YOQ-1>APMI0A,N3LLO-3,WIDE1*,WIDE2-1::N1YOQ-1 :EQNS.0,0.075,0,0,0,0,0,0,0,0,0,0,0,0,0

N1YOQ-1>APMI0A,N3LLO-3,WIDE1*,WIDE2-1::N1YOQ-1 :BITS.11111111,Telemetry test

Receiving stations which want to interpret telemetry values must remember the telemetry metadata in messages sent **to** a station and combine it with the raw data coming **from** the station.

Each analog channel can have three scaling coefficients in EQNS, which we will refer to as a, b, and c, here. For transmitted value, x, the final value is:

$$a\,x^2 + b\,x + c$$

In this case, telemetry data 174 * scaling factor 0.075 = 13.050 Volt. Full details for EQNS can be found in chapter 13 of the APRS specification found here .

There was not a PARM in this case, so there is no name for the value such as battery voltage.

Some implementations will simply send the number "13.05" rather than scaling it to an 8 bit integer.

2.4.3 Telemetry Rambling

Why limit it to only 5 analog channels, rather than making it more flexible? I suspect that the APRS Micro Interface Module (MIM) came first and the APRS specification followed that precedent. If anyone knows the real story, I'd like to know.

Does anyone know the history behind the metadata format? It would have been more obvious and simpler to use something like:

- T# for data
- TPARM for names/labels of values
- TUNIT for units
- TEQNS for scaling
- TBITS for bit sense (is 1 or 0 true)

If someone knows the history behind the decision to use more complicated special case of "**messages**" addressed **to** the telemetry station, please tell me.

There is another form of telemetry that appears in the Comment so more types of information can be conveyed by a single packet. It is described in the Comment section.

For more in depth discussion of APRS Telemetry, see
https://github.com/wb2osz/direwolf/blob/master/doc/APRS-Telemetry-Toolkit.pdf

2.5 Messages

The term “message” has a very specific meaning in APRS. It will cause confusion if you use that term to refer to something else.

- Position Reports are not “messages.”
- Object Reports are not “messages.”
- Status Reports are not “messages.”
- Weather Reports are not “messages.”
- Telemetry Data are not “messages.”

While most APRS transmissions are general broadcasts to everyone, it is also possible to direct a “message” to a particular destination.

2.5.1 Simple Case - RF only

If I wanted to send a message to N2GH the packet would look like this:

WB2OSZ-7>APK003::N2GH :Hi, Dave!{001

Broken into parts:

WB2OSZ-7>	Source Address.
APK003	Device Identifier.
:	End of addresses.
:	Data Type Indicator for Message.
N2GH	Addressee + enough spaces to make total of 9 characters.
:	Separator.
Hi, Dave!	Message Text. May include UTF-8.
{001	Message ID.

The optional Message Identifier has { followed by 1 to 5 alphanumeric characters. When a Message ID is included, the receiving station sends an acknowledgement that the message was received. Generally the sender will send several times until an ack is received or the maximum number of attempts is exceeded. The user should be notified whether the message was delivered successfully.

The other station responds with:

N2GH>APK003::WB2OSZ-7 :**ack**001

The acknowledgement is a special case of message where the content is lower case “ack” followed by the message id.

There is also a newer form of message identifiers with “}” in the middle, such as “{ab}cd”. Details might be included in a future revision.

2.5.2 More Interesting Case - with APRS-IS

When the addressee can't be reached over the local RF network, the packet can travel through an IGate station, across the APRS Internet Service (APRS-IS), and get retransmitted by another IGate station.

It is also possible to send messages to servers on the Internet. In this case, the "addressee" rules are relaxed.

Example: Send message to the "WHO-IS" server:

(1) WB2OSZ-7>APK003,WIDE1-1,WIDE2-1::WHO-IS :W1AW{0<0x0d>

(2) WB2OSZ-5>APDW17,WIDE1-1,WIDE2-1:}WHO-IS>APJIW4,TCPIP,WB2OSZ-5*::WB2OSZ-7 :ack0

(3) WB2OSZ-5>APDW17,WIDE1-1,WIDE2-1:}WHO-IS>APJIW4,TCPIP,WB2OSZ-5*::WB2OSZ-7 :C/ARRL HQ OPERATORS CLUB/CT/United States{1012

(4) WB2OSZ-7>APK003,WIDE1-1,WIDE2-1::WHO-IS :ack1012<0x0d>

There is a lot going on here.

First packet - original message from WB2OSZ-7 to WHO-IS:

WB2OSZ-7	Source of message.
APK003	Device id. You know how to look this up.
WIDE1-1,WIDE2-1	Typical digipeater via path.
:	":" means "APRS message."
WHO-IS :	"Addressee" -- Where the message is being sent to. The field must be exactly 9 characters and followed by ":" so Spaces might need to be inserted. Note that "IS" is letters "I" and "S", not fifteen.
W1AW	Message body -- A callsign.
{0	Message id 0, meaning an acknowledgement is expected. "{" followed by 1 to 5 alphanumeric characters.

Important point: The addressee is often the callsign and SSID of a radio station, but it doesn't need to be. The rules are more relaxed for the addressee because it could be more than 6 characters and/or have an alphanumeric SSID.

Next, we see an acknowledgement that WHO-IS received message id 0. The third party wrapper (crossed out below) needs to be removed by the receiving station before parsing the remainder.

~~WB2OSZ-5>APDW17,WIDE1-1,WIDE2-1:}~~WHO-IS>APJIW4,TCPIP,WB2OSZ-5*::WB2OSZ-7 :ack0

WHO-IS	Source of message.

APJIW4	System id.
TCPIP,WB2OSZ-5*	Added by IGate.
:	":" means APRS "message."
WB2OSZ-7 :	Addressee with a space to make 9 characters. Terminated with another ":".
ack0	Lower case "ack" is an acknowledgement for the following message id.

The WHO-IS server looks up the callsign and returns information. Again, the receiving station must remove the third-party wrapper before processing the "message."

~~WB2OSZ-5>APDW17,WIDE1-1,WIDE2-1:}~~WHO-IS>APJIW4,TCPIP,WB2OSZ-5*::WB2OSZ-7 :C/ARRL HQ OPERATORS CLUB/CT/United States{1012

WHO-IS	Source of message.
APJIW4	System Identifier = jAPRSIgate
TCPIP,WB2OSZ-5*	Added by IGate station.
:	":" means APRS "message."
WB2OSZ-7 :	Message "addressee," blank padded to 9 characters, then ":".
C/ARRL HQ OPERATORS CLUB/CT/United States	Response to query.
{1012	Message identifier.

Finally, the original station acknowledges the receipt of the information.

WB2OSZ-7>APK003,WIDE1-1,WIDE2-1::WHO-IS :ack1012<0x0d>

WB2OSZ-7	Source station.
APK003	Source device id.
WIDE1-1,WIDE2-1	Typical digipeater via path.
:	":" for "message."
WHO-IS :	Addressee, blank padded to 9 characters, and ":".
ack1012	Acknowledge message id 1012.

The astute reader will question why this part, with a data type indicator of "}"was left out of the explanation.

WB2OSZ-5>APDW17,WIDE1-1,WIDE2-1:**}**

That is the subject for our next section.

2.6 Third Party header

Third party headers are used to **transmit a packet on behalf of someone else**. I've only seen it used by IGate stations relaying a packet from APRS-IS to RF, but it is more general and could be used in other situations.

Perhaps this is best understood with an example. In the previous section, we saw how the station WHO-IS wants to send a message to WB2OSZ-7. The APRS-IS system knows that station has been heard recently by IGate WB2OSZ-7. APRS-IS forwards the message to this IGate.

It looks like this coming from APRS-IS:

> WHO-IS>APJIW4,TCPIP*,qAC,AE5PL-JF::WB2OSZ-7 :C/ARRL HQ OPERATORS CLUB/CT/United States{1012

First, the IGate station replaces the via path so it looks like this:

> WHO-IS>APJIW4,TCPIP,WB2OSZ-5*::WB2OSZ-7 :C/ARRL HQ OPERATORS CLUB/CT/United States{1012

The via path must contain **exactly**:

TCPIP	meaning it came from the Internet and must not go back again. An RF to APRS-IS IGate must not forward a packet with TCPIP in the path.
WB2OSZ-5	the IGate name.
*	both addresses have been "used."

It must be exactly like that for the whole system to work correctly. **No more. No less.**

That can't go directly on the air for two reasons:

- "WHO-IS" is not a valid AX.25 address because the SSID is letters.
- The source address field must be the station putting the packet on the air.

The solution is to wrap (or encapsulate) the desired packet with a third party packet. The result, sent over the air, is:

> WB2OSZ-5>APDW17,WIDE1-1,WIDE2-1:}WHO-IS>APJIW4,TCPIP,WB2OSZ-5*::WB2OSZ-7 :C/ARRL HQ OPERATORS CLUB/CT/United States{1012

The processing of this type of packet depends on the type of station that receives it.

(1) A **digipeater** only looks at the first via path (before the "}") and doesn't need to have any knowledge of third party packets.

(2) An **application** that wants to interpret the information part of the packet must first remove the encapsulation and process what is left over.

(3) An **RF-to-IS IGate** first removes the encapsulation and operates on the rest. If the remaining via path contains TCPIP, the packet must not be forwarded. This prevents loops.

It's a little more complicated but we will save that for another day. You might find this helpful to understand more about APRS Internet Gateways ("IGates").
https://github.com/wb2osz/direwolf-doc/blob/main/Successful-APRS-IGate-Operation.pdf

2.7 Weather

2.7.1 Complete Weather Report

Many hams have home weather stations and share the data over APRS. Example:

W1TG2>APU25N,UNCAN*:@091842z4256.20N/07049.42W_310/004g015t081r000p033P002h54b10001/ - Hampton, NH Wx<0x0d>

After the usual Source Address, Product Identifier, and digipeater path, we have:

@	Position Report with timestamp
091842z	Timestamp
4256.20N/07049.42W_	Same as other Position Report, but the "_" weather station symbol is a special case. Rather than a normal Comment, weather data is expected in a certain format.
310/004	Wind Direction (degrees clockwise from north), Wind Speed (knots)
g015t081r000p033P002h54b10001	Weather Data - same as wxnow.txt format produced by many home weather stations so conversions are not needed.
/ - Hampton, NH Wx<0x0d>	Comment

This is a special case of the Position Report when the symbol is "_" for weather station.

When the weather data is decoded, it comes out to be:
wind 4.6 mph, direction 310, gust 15, temperature 81, rain 0.00 in last hour, rain 0.33 in last 24 hours, rain 0.02 since midnight, humidity 54, barometer 29.54

2.7.2 Object Reports

Object Reports are much like Position Reports. The main difference is that you are sending information about some entity other than yourself.

A simple case would be advertising land marks, resources, or a local voice repeater so someone traveling through would know about it. If constructed properly, some APRS capable radios will allow you to set your voice frequency to information in the packet.

Besides having a single point location, objects can also cover regions defined by polygons.

Here is a more interesting, short duration, Object representing a Flash Flood warning during a heavy rain storm. It should be all one line but the page is not wide enough.

```
WZ0C-4>APN20H,W1MRA,WA1PLE-4*:}
GYXFFW>APRS,TCPIP,WZ0C-4*:
;GYXFFWNqA*140145z4443.50NF07012.30Ww FLASH_FLOOD
}j0IAdgJ`0T:5P5lAd{DNqAA<0x0d>

WZ0C-4>APN20H,W1MRA,WB2OSZ-5*:}BOXTOR>APRS,TCPIP,WZ0C-
4*:;BOXTORLtA*132230z4158.80N\07125.20Wt033/015 TORNADO
}a0IV,KOLQPTbObMcLcJdGV,{DLtAA<0x0d>
```

The first part is added by an APRS-IS to RF IGate station.
The rest is an object report.

The corresponding map display is in the next section.

2.7.3 NWS Bulletins

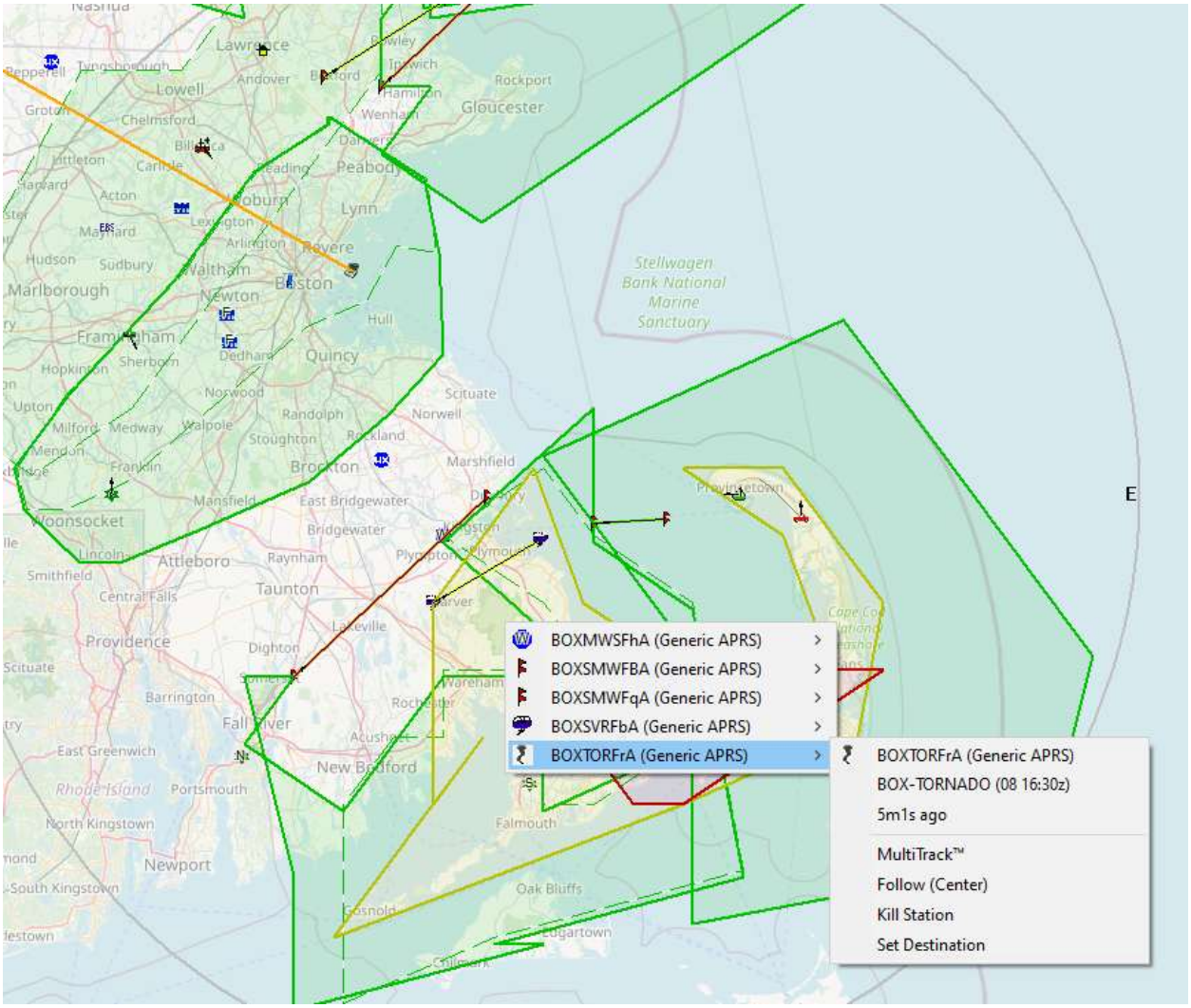
When the Addressee of a “Message” begins with “NWS” it is a National Weather Service Bulletin.

This is another special case of “messages” where the “addressee” is not a specific station. Receiving systems will usually have an option specifying what prefixes it wants to process in addition to the station name.

Example:

```
WZ0C-4>APN20H,W1MRA,WB2OSZ-5*:}BOXTOR>APRS,TCPIP,WZ0C-4*::NWS-WARN
:132230z,TORNADO,MAC005,MAC021,RIC007{DLtAA<0x0d>
```

Some applications, such as APRSISCE32 seen below, will display the regions impacted.



2.8 Queries and responses

Queries are used to ask other stations to reply with information. There are two formats:

- "**general**" to everyone, optionally within a specified circle, or
- "**directed**" to a specific station.

2.8.1 General Queries

These are intended for everyone, **over RF only**. IGate stations must not forward general queries from RF to APRS-IS.

Information part format:

?	Data type indicator.
type	See table below.
?	Trailing question mark

Optionally followed by target footprint:

lat	Latitude in degrees. e.g. 34.02 -- floating point degrees, not degrees/minutes.
,	comma
long	Longitude in degrees. e.g. -117.152 -- negative for west or south.
,	comma
radius	Radius in miles. e.g. 0200 because specification calls for four digits.

General Query Type	***Query***	***Response***
APRS	General — All stations query	Station's position and status
IGATE	General — Query all Internet Gateways	IGate station capabilities
WX	General — Query all weather stations	Weather report (and the station's position if it is not included in the Weather Report)

Query types must be in all upper case.

2.8.2 Directed Queries

These are sent as APRS "Messages" directed to a specific station. They never include a message identifier so there can't be an automatic message received ack. IGate stations handle these like any other message.

These are distinguished from other messages because the first character of the message text is a question mark.

Information part format:

:	Colon data type indicator
addr	Addressee, exactly 9 characters, trailing spaces if needed.
:	Colon after addressee.
?	First character of message text is question mark.
type	Directed Query Type, see table below.

Optionally followed by:

call Callsign of Heard Station - used by some query types.

Note that a question mark precedes all query types but only the general query is followed by a question mark.

Directed Query Type	***Query***	***Response***
APRSD	Directed — Query an individual station for stations heard direct	List of stations heard direct
APRSH	Directed — Query if an individual station has heard a particular station	Position of heard station as an APRS Object, plus heard statistics for the last 8 hours
APRSM	Directed — Query an individual station for outstanding unacknowledged or undelivered messages	All outstanding messages for the querying station
APRSO	Directed — Query an individual station for its Objects	Station's Objects
APRSP	Directed — Query an individual station for its position	Station's position
APRSS	Directed — Query an individual station for its status	Station's status
APRST or **PING?**	Directed — Query an individual station for a trace (i.e. path by which the packet was heard)	Route trace

3 Comments

Position Reports (including MIC-E) and Object Reports can have free form text comments.

APRS was not designed to add new attributes in a consistent way. As a result, we see numerous hacks where character strings, with certain patterns, have special meanings.

3.1 cse/spd - Course and Speed

cse/spd

The course is degrees (001-360) clockwise from north.
Speed is in knots. Three digits, leading zeros as needed.
Zeros, spaces, or periods can be used if values are not known or relevant.

If used, this must appear immediately after the location and symbol, before any frequency or general comments. Technically it is called a data extension.

3.2 Power, Height, Gain (PHG)

Transmitter power, combined with antenna height and gain, can give a very rough approximation of range.

It is composed of the 3 upper case letters "PHG" and 4 digits.

```
!DDMM.mmN/DDDMM.mmW#PHG5360 …comments…
  |          |       | | ||||
  |          |       | | ||||_________ Omni (Direction of max gain)
  |          |       | | |||__________ Ant gain in dBi
  |          |       | | ||___________ Height = log2(HAAT/10)
 LAT       LONG      | | |____________ Power = SQR(P)
                     | |_____________ Power-Height-Gain identifier *
                     |______________ # is symbol for digipeater
```

There are only 10 possible values for each of these fields as follows:

Digit	0	1	2	3	4	5	6	7	8	9		equation
Power	0	1	4	9	16	25	36	49	64	81	watts	sqrt (P)
Height	10	20	40	80	160	320	640	1280	2560	5120	feet	log2(H/10)
Gain	0	1	2	3	4	5	6	7	8	9	dBi	
Direction	Omni	25	90	135	180	225	270	315	360	-	degrees	D/45

Height is above average terrain (HAAT), not above sea level.
Direction must always be specified, even if it is omnidirectional (0).

If used, this must appear immediately after the location and symbol, before any frequency or general comments. Technically it is called a data extension.

Since course/speed and PHG are both data extensions, using both in the same packet is not allowed. Just because it is invalid, that doesn't stop people from doing it.

Occasionally, you might see an extra digit followed by a "/" character.

> N8VIM>BEACON,N3LLO-3,W1MHL*,WIDE2:!4240.85N/07133.99W_PHG72604/ Pepperell, MA. WX. 442.9+ PL100<0x0d>

The extra digit represents the number of beacons transmitted per hour, 4 in this example. This can be used to calculate how reliably that station can be heard.

This hack, which was added later, violates the rule that data extensions must be 7 characters long.

3.3 Voice Frequency

People will often want to advertise the voice frequency where they are listening. APRS-equipped radios usually have a TUNE or QSY button that will switch the voice frequency to the frequency specified in the selected packet. It is imperative that precise formatting rules are followed so the radios can parse the frequency specification.

This is just a simplified explanation which covers the most common cases. The APRS Protocol Specification version 1.2, chapter, 18 has all the details for all the less common cases.

This must appear at the beginning of the free form part of the comment text, after any course/speed or PHG. The form must appear exactly as follows.

Frequency:	999	3 digits representing MHz.
	.	Decimal point.
	999	3 digits representing kHz.
	MHz	Exactly those letters. **Case Sensitive**. Not MHZ or mHZ or …

This can optionally be followed by a CTCSS tone and/or transmit offset.

Tone:	(space)	Space separator.
	T	Upper case letter T.
	999	Integer part of tone. Leading 0 if needed to make 3 digits.
TX offset:	(space)	Space separator.
	+ or -	Plus or minus.
	999	Offset in tens of KHz so 600 kHz would be "060".

Another space should be added if there is anything else in the comment field.

Here are a few people who know how to do it correctly.

W1STJ-9>T2TU4Q,N1SFT,WIDE1,UNCAN,WIDE2*:`c8um^9j/`"4I}146.685MHz T100 -060_1

KB1TOY-9>TRRY9U,W1MHL*,WIDE2-1:`c_"l <0x1c>j/`449.075MHz T088 -500_%

W1GBH>TRRQ5Z,WA1PLE-4,WIDE1*:`ca0l7Wj/`"3r}146.655MHz T088 -060_%

N1EZ-7>P0PPPP,N3LLO-3,WIDE1*,WIDE2-1:'vX<0x1c>l <0x1c>[/>"3r}146.685MHz T100 -060^

Actually, these are all MIC-E format, from Yaesu and Kenwood. It is quite likely that the radio automatically generated the proper format.

For a voice repeater Object Report, the frequency can be in the first 7 characters of the object name to make it stand out more. The other 2 characters should be chosen to make the name unique. It really doesn't matter if there are global conflicts because this is locally useful information you would get over the radio.

EKONCT>BEACON:;146.730CT*111111z4134.84N/07206.31Wr146.730MHz T156 R30m ECTN 9P DAILY RASON

In this case we have "R30m" meaning the repeater has a range of about 30 miles. I don't know if it is still true, but at one time Yaesu radios could not parse the frequency in the object name. **The frequency should also appear in the comment field for greatest compatibility.**

These are only the most common cases. The complete specification can be found in chapter 18 of the APRS protocol specification version 1.2 found here.

3.4 Altitude

An optional altitude can be added anywhere in the comment field. It must have exactly this format with upper case "A" and six digits:

```
/A=123456
```

The number is in feet (0.3048 meter) above mean sea level. Applications should convert to local units for display.

There is a surprisingly large number of places on the Earth's surface that are below sea level. The MIC-E format allows a negative altitude but the /A= format does not allow that.

Many applications have added this obvious trivial enhancement. The altitude can also be expressed as a minus sign and exactly five digits to maintain a consistent total width.

```
/A=-12345
```

If you encounter an application that does not recognize this form, please encourage the author to add this simple enhancement. If an APRS Working Group is ever resurrected, we should try to get this into the official standard.

3.5 DAO - Increased Resolution

APRS latidude and longitude coordinates have the resolution of 1/00 of a minute. This on the order of 10 meters on the ground. You can add two more digits of precision xxxxxxxxxxxxx

N83MZ>T2TQ5U,WA1PLE-4*:`c.l+@&'/'"G:} KJ6TMS|!:&0'p|!w#f!|3
MIC-E, Small Aircraft (original primary symbol), Byonics TinyTrack3, In Service
N 42 41.5502, W 071 18.8076, 283 km/h (176 MPH), course 210, alt 1764 m (5787 ft)
Seq=25, A1=470, A2=625
KJ6TMS

Broken down:

N83MZ	Original sender of packet. That is not a ham callsign. It turns out to be an aircraft registration.
T2TQ5U	Latitude 42° 41.55 (note two decimal places)
WA1PLE-4*	We heard this digipeater.
`c.l+@&'/'	Longitude 71° 18.80 (note two decimal places) Course 210, speed 283 km/h.
"G:}	Altitude 1764 meters.
KJ6TMS	Comment with ham call sign of sender.
|!:&0'p|	Base-91 telemetry (see next section). This should be after the free text part of the comment.
!w#f!	Add 0.0002 to latitude. Add 0.0076 to longitude. This should be after the free text part of the comment.
|3	TinyTrack 3

"!" 3 characters "!" provides about two more digits of location precision, giving resolution on the order of 100 cm.

This is described in chapter 5 of the APRS protocol specification version 1.2 found here.

3.6 Base-91 telemetry

Using the same example as the previous section, let's focus in on this:

|!:&0'p| Base-91 telemetry.

Rather than having a separate Telemetry Data packet, it can be embedded after the free text part of a comment. The format is:

| two-thru-seven pairs of characters |

Each pair of characters represents a number in base-91 notation. Decimal values of 0 through 8280 can be represented. The first is the sequence number. If a seventh pair is present, the number represents 8 values of one bit each.

In this example, we get :

- Sequence number = 25
- Analog value 1 = 470
- Analog value 2 = 625

This is described in chapter 13 of the APRS protocol specification version 1.2 found here.

3.7 !x! Means Do Not Archive

Any packet containing this string in the comment or message text field should not be archived by any of the APRS-IS databases. It won't show up on aprs.fi. It applies to positions, objects, status, and messages.

The x is a literal lower case x, not a placeholder for any alphanumeric character.

3.8 UTF-8 Characters

Originally, APRS was created to use only the 7-bit ASCII character set. As it gained more international usage, it became painfully obvious that needed more letters than the Latin alphabet. Other special characters, such as the degree sign are very useful.

Most of the modern APRS applications handle UTF-8 properly. Here is an example of how I'm trying to raise awareness and find cases where UTF-8 is not handled properly.

> WB2OSZ-5>APDW17:!4237.14NS07120.83W#PHG7140 Did you know that APRS comments and messages can contain UTF-8 characters? アマチュア無線

4 Examples of Errors Seen

Unfortunately, it's not uncommon to see improperly constructed packets. Sometimes this is due to defective software or improper configuration. More often, beginners, without proper mentoring, don't know the proper way to do things. The official APRS documentation is scattered so it is not always easy to find an answer.

direwolf will often detect errors and provide an explanation. My intention is not to pick on certain people; I'm just using these real world examples as teaching examples so others can learn and avoid the same mistakes.

If you see others making these mistakes, be a good "Elmer" and help them improve their operating skills.

4.1 Non-APRS format packets on the APRS frequency

Sometimes people throw random stuff out there without even trying to make it look like APRS. The first character of the information part should be the APRS data type indicator.

K1EQX-7>APMI03,N1LIT-6,N3LLO-3,WIDE2*:NFMRA// K2LM@nycap.rr.com<0x20>

K1FFK>APMI03,W1MRA,UNCAN,WIDE2*:NOBARC.org // K2LM@nycap.rr.com // N1ATP.com

W1IMD>BEACON,KQ1L-8,AB1OC-10,WIDE2*:W1IMD HIRAM, ME<0x0d>

W2AIQ-1>BEACON,EKONCT,N3LLO-3,WIDE2*:Bi-Directional-I-Gate CODE Enhanced-Re-Tooled for Speed

WA2GUG-15>ID,K1FFK,N3LLO-3,WIDE2*,WIDE1-1:WA2GUG-15/R DISABL/D *-1/B<0x0d>

W3TWA-1>T0TW3W,EKONCT,W1MV-1,WIDE2*:w3twa-3 digi<0x0d>

W1DMR-4>APMI06,WA1PLE-4*:PHG3530/APRS Domination

Where is this BBS so an antenna can be pointed? What packet frequency is it on?

WZ0C-4>ID,KB1EMU-10,WIDE1,W1MHL*,WIDE2:WZ0C-4 BBS [MAYNARD, MA]

This one is not much of a tracker if it does not send a location.

N1HRK-2>BEACON,K1RK-1,WA1PLE-4*:KPC3+ TRACKER N1HRK@ARRL.NET<0x0d>

It seems, to me, that if you are transmitting AX.25 UI frames on the APRS frequency, you should make at least some attempt to construct a valid APRS packet. In general, if you are talking about yourself, use a Position Report. If talking about something else, use an Object Report.

4.2 Missing System Type Identifier

With the exception of MIC-E format, the destination field should contain the system identifier of the form AP*xxxx* to indicate the type of device or application that generated the packet. When you see "APRS" in documentation examples, it is just a placeholder for the actual value. We don't know the type of software being used for these stations.

> KR2C-1>APRS,WA1PLE-4*,WIDE2-1:}KK7MGJ-7>APWW11,TCPIP,KR2C-1*::W2ILT :N:HOTG Greetings from AZ!

> W1YK-1>APRS,WIDE:!4216.47B/07148.43W#PHG5350 W2, WIDE1-1, WPIWA<0x0d>

Here is another case. The packet was digipeated five times. APRS-IS forwards messages to IGate stations only if the IGate heard the message addressee over the radio. There is no reason to use an extreme digi via path. The addressee is not that far away.

> N2MH-15>APRS,KD2CIF-1,KC2OUR-1,K1FFK-1,N3LLO-3,WB2OSZ-5*:}WLNK-1>APWLK,TCPIP,N2MH-15*::KD9BBB :You have 1 Winlink mail messages pending{4496

Using BEACON is wasting air time without conveying useful information. If it is experimental, APZ should be used.

> NE1CU-10>BEACON,KB1AEV-15,N3LLO-3,WIDE2*:@221226z4114.44N/07300.72WrMilford CT. PS=12.5V, Shack=85.2F

These have empty destination (device id) fields. How could that happen?!?!?

> KB1EZZ-9>,W1IMD,UNCAN,WIDE2*:!4413.87N\06936.24Wc205/041/A=000093EMA 902 COMMAND POST

> VE9FPG-2>,W1LH-9,KQ1L-1,UNCAN,WIDE3*:!4612.01NS06710.74W#PHG5460/W3 CRABBE MTM,NB MARCAN UIDIGI

> W1BRI-7>,W1MRA*,WIDE2-1:!4217.68N/07130.31W&267/000/A=000204W1BRI vai AT_D878UV PLUS

I wrote to one of them and he is using an Anytone radio. **The manufacturer should hardcode the destination field with the assigned system type identifier. Instead Anytone allows the user to set this.** Documentation is nearly non-existent so newcomers to APRS would not know what to put there.

Here someone put a callsign in the destination field. Why? Based on the comment it looks like it might be another Anytone radio.

> KC2DSH-9>N2MH-15,EKONCT,N3LLO-3,WIDE2*:!4041.10N/07428.38W[274/001/A=000132KC2DSH-Anytone-APRS

WIDE1-1 should be in the digipeater via path, not in the destination field.

> WA2NAN>WIDE1-1,VE3PGC,VE2PCQ-3,WIDE2,MTWASH,N3LLO-3*,WIDE2-1:;WA2NAN-1 *062019z4414.41N/07505.66W#FINE, N.Y. DIGI<0x0d>

This also has an abusive digipeater path. The packet already went through at least 4 digipeaters and still has an unused address. RF users don't care about a digipeater 350 miles (560 km) away. This is just unnecessarily clogging up the network.

The UISS application gives the user a choice of CQ, APRS, BEACON, QST, APRSAT, or custom. The assigned system identifier

Some MIC-E position reports are missing the device identifier. What are they?

> VE2VL-9>TU3U0X,VE2RAW-3,W1UWS-1,W1MRA*,WIDE2:`eF%o\9v/"41}
>
> N1OHZ>T2QT2T,W1MRA*,WIDE2-1:'cN]l <0x1c>-/
>
> W8BAP-1>S9QS3U,KD8DNS-1,K8GPS-4,MAYVIL,CASCTY,N3TJJ-13,N3TJJ-7,AD3O-3,W1MRA*:`nVF<0x1c> <0x1c>#/ repeaters 146.85- PL74.4<0x20>

The last one has an abusive path. The packet was digipeated 8 times before I heard it. We don't care about your voice repeater 820 miles (1320 km) from here.

4.3 Unregistered System Type Identifier

Many stations around here have device/system identifier APN000 which is not assigned in the database.

> W1FSH-9>APN000:!4234.38N<0x00>07144.77W<0x00>116/000
>
> N2RJ-9>APN000,MATWAN,WIDE1,KB1AEV-15,N3LLO-3,WIDE2*:!4054.45N/07423.84W>154/000
>
> KC1OCA-6>APN000,KB1AEV-15,N3LLO-3,WIDE2*:!4208.87N/07226.34W[092/000

KB5LNC-6>APN000,KV3B-2,WIDE1,CHATSW,K1RK-1,WIDE2*:!3852.04N/07703.46Wk321/000

W1NIG-1>APN000,W1MHL*,WIDE2-1:!4211.46N/07119.34Wk360/000

KN0O-1>APN000,WA1PLE-4*,WIDE2-1:=4409.52N/06907.06W-123/000VGC beacon

N1DDH-10>APN000,N3LLO-3,WIDE1*,WIDE2-2:!4252.57N/07127.70W-

KF1D-9>APN000,W1MRA*,WIDE2-1:=4204.05N/07128.54W>195/019Bryan, KF1D Mobile 13.27V

KC1OCY-9>APN000,WA1PLE-13*,WIDE2-1:=4219.28N/07107.25W>242/022146.520MHZ winlink

KC1PYM-9>APN000,W1MHL*,WIDE2-1:=4215.25N\07056.21Wk034/012Vero VR-N7500 14.05V

AF1SL-9>APN000,W1XM*,WIDE2-1:=4215.13N\07056.30Wk170/022Vero VR-N7500 14.31V

The last two have Vero VR-N7500 in the comment. Are all of these using that type of radio or is the same system identifier being used by multiple products or even multiple manufacturers?

Update: The BTECH UV-Pro used APN000 but now uses the registered device id of APBTUV.

4.4 Obsolete Digipeater Form

Most often the initial digipeater field uses generic digipeater addresses like WIDEn-n. Notice that just "WIDE" is used here:

W1YK-1>APRS,WIDE:!4216.47B/07148.43W#PHG5350 W2, WIDE1-1, WPIWA<0x0d>

KQ1L-1>APNU19,WIDE:!4414.82NN07025.15W#PHG5730/KQ1L-1 Streaked Mountain, Buckfield, Me.

K2TGX>APW275,W1MHL*,WIDE:=4156.43N/07111.75WyPHG5260/WinAPRS 2.7.5 - MABRINORTON -275-<630><0x0d>

N1IQI>WIDE,W1MV-1*,WIDE:=4202.59N/07050.08WNrfn Pembroke,ma NTS {UIV32N}<0x0d>

Using simply "WIDE" without any numbers has been obsolete for about 20 years. Unfortunately there are still many websites, which haven't been updated for decades, offering bad advice. Newcomers are told to do something that won't work.

4.5 Empty Digipeater Name in Path

This is really strange. In monitoring format, there is a comma after the system identifier and then colon. That means there was a digipeater specified, in the AX.25 format, but its name was blank.

W1BKW-4>APNU19,:!4414.97NN06918.50W#PHG5730 W1BKW-4 Coggins Hill, Union, ME

This appears to be a replacement EPROM for the TNC-2 last updated in the year 2004.

4.6 Invalid Location

K2VUD-1>APK102,WA1PLE-13*,WIDE2-1:=09H6.00N/134E9.00p_306/001g t025r000p000P h55b10249KDvs<0x0d>

Let's break it down:

=	Position Report.
09H6.00	We were expecting: 2 digits for degrees latitude. 2 digits for minutes. **Where did "H6" come from?** Decimal point. 2 digits for hundredths of minutes.
N	Hemisphere.
/	
134E9.00	We were expecting: 3 digits for degrees longtitude. 2 digits for minutes. **Where did "E9" come from?** Decimal point. 2 digits for hundredths of minutes.
p	Expecting E or W for hemisphere. **Where did "p" come from?**
_	Symbol for weather.
306…..	Weather data.

Here is an example which was probably hand coded rather than generated by an application.

W1YK-1>APRS,WIDE:!4216.47B/07148.43W#PHG5350 W2, WIDE1-1, WPIWA<0x0d>

Here we find "B" rather than the expected "N" for hemisphere so the location is not valid. This was most likely a typographical error. Letters B and N are adjacent on the keyboard.

4216.47B

4.7 Incorrect Timestamp in Position Report

I didn't mention this earlier because I think this is the first time I've seen it used. There is a rarely used variation on the Position Report where it contains a timestamp. This would be useful for storing positions of a moving thing and transmitting them later rather than real-time. There are three different formats which can be distinguished by looking at the 7th character.

- *ddhhmm***z** -- Zulu (GMT, UTC) -- recommended
  - dd = day of month
  - hh = hours
  - mm = minutes
  - z = literally lower case z

- *ddhhmm***/** -- Local time
  - dd = day of month
  - hh = hours
  - mm = minutes
  - / = slash character

- *hhmmss***h** -- Zulu
  - hh = hours
  - mm = minutes
  - ss = seconds
  - h = literally lower case h

This does not match any of the acceptable timestamp formats. In this case it is only 4 digits before the z.

K9WK>APMI0A,CHATSW,WIDE1,K1RK-1,WA1PLE-4*:/0000z3946.09N/07529.71W>K9WK Station.

This one has two underscores in place of the day of month. That is not allowed.

WZ0C-4>APN20H,W1MRA*,WIDE2-1:}AA1HO>API510,TCPIP,WZ0C-4*:/__1552z4238.34N/07119.94Wv/<0x0d>

A Position Report timestamp must be 6 digits followed by z, h, or /. This is explained in Chapter 6 of the APRS protocol specification.

4.8 APRS is case-sensitive

We have two different issues here:

N1EOE>APN391,N1NCI-3*,WIDE2-1:!4216.95n/07243.20w#phg6230/ Easthampton MA<0x0d>

- The hemispheres are lower case n and w. These must be upper case.
- PHG must also be upper case so this is treated like part of the free form text.

4.9 Kenwood Bug - 0xFF bytes

Sometimes, apparently at random, the Kenwood TM-D710 will insert a bunch of 0xff characters on the packet.

W1SHS-9>4R1X9U,W1MRA,WB2OSZ-
5*:`c0<0x1d>mIL>/]"4T}<0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff>
<0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0x
ff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff>=<0x0d>

VA2RN-9>T3PS0Q,KA2QYE-10,WIDE1,K1EQX-7,UNCAN,WIDE3*:`eJ1!"&>/]"4h}<0xff><0xff><
0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff
><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0
xff><0xff><0xff><0xff><0xff>=<0x0d>

WS1EC-1>TSTS8S,KA1GJU-3,WIDE1,KB1TSO*:'b5-l <0x1c>-
/]449.225MHz<0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><
0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff
><0xff><0xff><0xff><0xff>=<0x0d>

N2KI-9>T1TT7V,K2RVW,WIDE1,K1FFK,N3LLO-3,WIDE2*:`f+: s4k/]"7w}147.390MHz T156 +0
60<0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><
0xff><0xff><0xff><0xff>=<0x0d>

K1MGR-9>T3SX1W,W1IMD,WIDE1,UNCAN,WIDE2*:`b2ol <0x1c>j/]"4;}147.090MHz C100
+060<0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff
><0xff><0xff><0xff><0xff>=<0x0d>

K1DSP-9>TSRY7W,K1EQX-7,WIDE1,N3LLO-
3,WIDE2*:`eDao^%>/]"5"}147.730MHz<0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0
xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff>
<0xff><0xff><0xff><0xff>=<0x0d>

KE2X-9>T0TX6V,N2ACF-3,WIDE1,KB1AEV-15,N3LLO-3,WIDE2*:`eK[og`j/]"4+}147.390MHz C156
+060<0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff
><0xff><0xff><0xff><0xff>=<0x0d>

W1HS-4>T34S3S,W1UWS-1,WIDE1,N3LLO-3,WIDE2*:'d+mlH<0x1c>R/]"5`}/steve - RV Mobile TM-D710<0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff>=<0x0d>

These final examples are more extreme with 0x00 and 0x0f bytes mixed in too.

KC2ASA-9>TR5Q3T,K1EQX-7,WIDE1,N3LLO-3,WIDE2*:4P<0x00><0x0f>4T<0x00><0x0f>4X<0x00><0x0f>4\<0x00>`eMQl<0x1c>>/]"3r}445.825MHz T114 -500<0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff>=<0x0d>

KB1HNZ-9>TSSP5X,W1IMD,WIDE1,KQ1L-8,N3LLO-3,WIDE2*:4P<0x00><0x0f>4T<0x00><0x0f>4X<0x00><0x0f>4\<0x00>`b64l!p>/]"4:}449.225MHz<0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff><0xff>=<0x0d>

Other TM-D710 stations to not exhibit this strange behavior. It would be interesting to find out what version firmware is being used and whether updating clears up the problem.

4.10 Kenwood Bug - Inappropriate Message Reject

Here we have a weather bulletin:

WZ0C-4>APN20H,WA1PLE-4,WIDE1*,WIDE2-1:}BOXMWW>APRS,TCPIP,WZ0C-4*::NWS-WARN :091215z,Storm Warning,ANZ236{100AA<0x0d>

Three stations respond with a message reject.

K2VUD-1>APK102,WA1PLE-4,WIDE1*,WIDE2-1::BOXMWW   :rej3<0x0d>

AB1OC-10>APK102,WIDE1-1,WIDE3-3::BOXMWW   :rej3<0x0d>

WA1PLE-4>APK102,WIDE2-1::BOXMWW   :rej3<0x0d>

"rej" means the receiving station is unable to receive the message. For instance, an APRS-equipped radio might have a limited amount of incoming message memory and it is full.

This is wrong because the "ack" or "rej" reply should happen ONLY for a "message" addressed to that station. They should never be sent in response to a bulletin.

Where did they get the message id 3 from? Why not 100AA?

4.11 Kenwood Bug - Inappropriate Automatic Reply

Kenwood has a feature which allows you to send an automatic replay (an addition to any ACK) to messages addressed to your station. It is possible to configure responses to only specific stations or to a wild carded name.

Bob WB4APR had this to say about it, "Auto Reply is only for passing along special information to anticipated message senders when the operator may temporarily not be able to respond. It should not be left on for routine operations in most cases because it adds unnecessary network load. The sending station will always get an ACK, and this is sufficient to know that the message was delivered in most cases."

Consider this observed example: BOXMWW and N1SFT send bulletins:

> WZ0C-4>APN20H,W1MRA,WA1PLE-4*:}BOXMWW>APRS,TCPIP,WZ0C-4*::NWS-WARN :091215z,Storm Warning,ANZ236{100AA<0x0d>
>
> WZ0C-4>APN20H,W1MRA*,WIDE2-1:}N1SFT>APWW11,TCPIP,WZ0C-4*::BLN0 :NEW ENGL. FUSION GROUP TECHNET SUNDAYS 3PM, WIRES-X RM 28941<0x0d>

Kenwood radios incorrectly send an auto reply to bulletins. Imagine if every station did this in response to bulletins!

> AB1OC-10>APK102,WIDE1-1,WIDE3-3::BOXMWW :AA:Message Recvd. by AB1OC-10<0x0d>
>
> AB1OC-10>APK102,WIDE1-1,WIDE3-3::N1SFT :AA:Message Recvd. by AB1OC-10<0x0d>

Automatic replies should only be sent when messages are addressed to the specific station.

In his musings, Bob WB4APR concluded, "Auto-Answer messages are SPAM to the network in most cases."

4.12 Kenwood Bug – Adding carriage return after a message id.

A "message" is sent with a message id of 84:

> KZ2X-5>APDW17,N3LLO-3,WIDE1*:}N1IQI>APU25N,TCPIP,KZ2X-5*::W1JT-7 :Merry Christmas{84

When a Kenwood radio sends an acknowledgement, it appends a carriage return:

> W1JT-7>APK102,AJ1L,W1MHL,N3LLO-3,WIDE2*::N1IQI :ack84<0x0d>

The protocol specification says nothing about adding a carriage return at the end. The required format is:

- “ack”
- Message id
- End of packet

The sending station is expecting to see “ack84” so it knows to stop sending additional copies of message 84. Instead it gets “ack84<0x0d>” which the sender might or might not match depending on whether the application is aware of the Kenwood bug and puts in a hack to work around it.

4.13 Incorrect Frequency Specification

Sometimes you might want to advertise that you are listening to some voice frequency. Many will put this into the comment field. Most people do it wrong, probably because they are unaware of the standard. We covered the correct format earlier. All of these are wrong.

K1ASM-9>EB2Q7Z,EKONCT,N3LLO-3,WIDE2*:`d_<0x1c>m6^k/]"47}[scanning]Monitoring 146.520=

K1RBC-9>T3TT5U,W1IMD,WIDE1,UNCAN,WIDE2*:`c\<0x1e>n|\>/`"5q}Monitoring 146.520_%<0x0d>

K1RTA-3>TQUV2X,W1SGL-2,WA1PLE-4*:`bYBl <0x1c>k/`"4%}Russ on 146.520 in a 14 Ram and FTM-400XDR_%<0x0d>

K5HIP-7>TRQP8Z,K2RVW,WIDE1,K1FFK,W1MRA*,WIDE2:`e1rm_D>/`"66}listening 146.52_0<0x0d>

KC1DDH-9>TRQS7U,W1MRA*:`c;(l}Dj/`144.390 PL100_%<0x0d>

KC1HHK-9>T2QS5S,WA1PLE-4,WIDE1*:`cYvm!EE/`"5n}KC1HHK - PAUL 146.67_%<0x0d>

KE5BM-9>TQSY7S,W1SGL-2,W1MHL,N3LLO-3,WIDE2*:`b(.m,&>/`"3r}Black VW Passat 146.520 & VA_1<0x0d>

N1EZ-1>APWW11,AB1OC-10,WIDE1,W1MRA*,WIDE2:@021909h4255.25N/07134.38Wl146.685 in Shack<0x0d>

N1NW>T1ST8T,EKONCT,W1MRA,N3LLO-3,WIDE2*:'d^9l <0x1c>#/]N1NW 146.730 TONE 156.7<0x0d>

K1RV-9>TR1P3W,AJ1L,W1MHL*,WIDE2:`bV8n?V>/`"4F}Monitoring 146.535 Simplex_"<0x0d>

K3JDG-7>APAT81,K1EQX-1,WIDE1,N3LLO-3,WIDE2*:!4316.57N/07343.27W[205/000/A=000246146.730

This one is close, but there should be a space after each of the frequency, tone, and offset. **Do not** put a space in front of the frequency.

W1BST>APTT4,W1IMD,WIDE2,UNCAN*,NH3-1:!4341.89NL07109.20W#PHG3660147.030MHzT088+060EL#875273 77F 13.7V

Refer to earlier section on detailed explanation.

This is very important because some APRS-equipped radios will parse the frequency, offset, tone specification, in the selected position or object report, and allow you to set your voice frequency with the touch of a button. Some applications will parse this information and present it to the user.

Improperly formatted data breaks that feature.

4.14 Incorrect Power, antenna Height & Gain

PHG is pretty easy but I've still seen a few people mess it up. There are 3 things to remember:

The protocol specification clearly states that any optional PHG is specified, it should appear at the very beginning of the comment, before the free text part, for the greatest compatibility. Some applications might recognize it anywhere, but officially it needs to be first.

Some applications might recognize PHG in the middle of the comment, but the APRS Specification clearly states it must be immediately after the location and symbol. An application written with strict adherence to the standard would treat PHG as part of the free form comment text.

UNCAN>APOT30:!4258.99N/07135.29W# 10.8V 98F PHG37306/ N1PA-Mt Uncanoonuc Digi

It should be rearranged like this to adhere to the standard:

UNCAN>APOT30:!4258.99N/07135.29W#PHG37306/ 10.8V 98F N1PA-Mt Uncanoonuc Digi

This one has only 3 digits after PHG. This is not valid. It also appears after course/speed so, even if it had the correct format, it might not be recognized by some applications.

KE1IU-9>APTT4,WB2OSZ-5*,WIDE2-1:/152720h4236.54N/07118.94W>251/059/PHG404/KE1IUMark@gmail.com

4.15 Wrong character code for degree symbol

Here are two examples of using the wrong character code for the degree symbol.

> W1TG-1>APU25N,WA1PLE-4,W1MRA*,WIDE2:>232322zDX: W1SGL-2 41.41.93N 70.18.20W 89.5 miles 162<0xf8> 19:14<0x0d>
>
> KG5KTN-1>APWW11,W1WQM,WIDE1,N3LLO-3,WIDE2*:>FN42kw/-DX: KQ1L-8 28.7mi 48<0x0b0> 01:23 4313.42N 07041.56W<0x20>

ASCII does not include a degree symbol. Different people came up with their own (incorrect) work-arounds. The first example used 0xf8, from Microsoft code page 437, and the second used 0xb0, from ISO Latin 1. Both are wrong.

When using a character which is not in ASCII, the correct approach is to use UTF-8 encoding. The correct encoding for the degree symbol ° is the two byte sequence **0xc2 0xb0**.

4.16 DX Report missing mandatory space

Here there should be a space between "/-" and "DX".

> KG5KTN-1>APWW11,W1WQM,WIDE1,N3LLO-3,WIDE2*:>FN42kw/-DX: KQ1L-8 28.7mi 48<0x0b0> 01:23 4313.42N 07041.56W<0x20>

A status report, with grid square, requires a mandatory space after the symbol table and symbol code "/-" in this case. The reasoning behind it is not clear, but that is what the protocol spec requires. Applications that process this type of packet might not be able to parse it.

4.17 Incorrect Queries

Queries are used to ask other stations to reply with information. There are two formats:

- "general" to everyone, optionally within a specified circle, or
- "directed" to a specific station.

APRSD means "What stations have you heard?"

> N1OLA>APAGW,K1EQX-7,W1UWS-1.N1NCI-3,WIDE1,W1MRA*,WIDE2-1,WIDE3-2:?APRSD

This packet has three issues:

(1) This is the general query format, not directed toward a specific station. With that extreme path, it would go really really far and possibly generate a massive number of responses.
(2) "APRSD" is only allowed for "directed" queries, not general queries.
(3) The protocol specification clearly states that the **general** query form must have a question mark after the query type. A proper implementation would not respond to this for both reasons (2) and (3).

This one:

N1OLA>APWW11,K1EQX-7,N3LLO-3,WIDE1*::VE2PCQ-3 :?aprsp

is incorrect because the query type must in in upper case.

This one:

KE2BSD-7>APY03D,W2AEE,N2ACF-3,WIDE1,KB1AEV-15,N3LLO-3,WIDE2*::KE2BSD-15:?APRSP{25

is incorrect because a directed query must not have a message id.

4.18 Obsolete Weather Format

This has a vendor-specific format rather than the standard format.

N8VIM>APN391,AB1OC-10*,WIDE2-1:$ULTW00A2007C0317012E27CFFFFA89AB000101B300EB034300000075<0x0d><0x0a>

Raw Weather Formats are not recommended. The sending system should convert to the standard complete weather format.

4.19 Used Digipeater Address Not Marked as Used

Notice the progression here.

KB1TSO>APDW16,WIDE1-1,WIDE2-1:!4242.77NS07113.26W#PHG7150Methuen, MA DIGI
KB1TSO>APDW16,WA1PLE-13*,WIDE2-1:!4242.77NS07113.26W#PHG7150Methuen, MA DIGI
KB1TSO>APDW16,WA1PLE-13,W1MRA*,WIDE2:!4242.77NS07113.26W#PHG7150Methuen, MA DIGI

- WIDE1-1 is changed to WA1PLE-13*. This is correct.

- WIDE2-1 is changed to W1MRA*,WIDE2. Wrong. WIDE2 is not marked as used. Better yet would be to remove it.

Here is another case of the same thing:

K1RV-9>TR1P3W,AJ1L,W1MHL*,WIDE2:`bV8n?V>/`"4F}Monitoring 146.535 Simplex_"<0x0d>

Both problematic digipeaters are running a Kantronics KPC-3+ with 8.2 firmware. (device id APN382) According to multiple sources, such as https://www.nwaprs.info/widen-n.html it has this bug. See part near end highlighted in yellow.

> *The bug causes the TNC to insert it's callsign has-been-digipeated bit set on the name of the digipeater instead of on the used up WIDE1 path.*

We also see where the Microsat (APMI03 & APMI06) also fails to mark a used digipeater address as used. Did they copy this bug from Kantronics?

W8BAP-1>S9QS3U,KD8DNS-1,K8GPS-4,WOOSTR,N3DXC-2,KD2CIF-1,KC2OUR-1,K1FFK*,WIDE1:`nVF<0x1c> <0x1c>#/ repeaters 146.85- PL74.4<0x20>

W8BAP-1>S9QS3U,KD8DNS-1,N8CUB-4,K8GPS-4,K8GPS-10,WOOSTR,K1FFK,N3LLO-3*,WIDE1:`nVF<0x1c> <0x1c>#/ repeaters 146.85- PL74.4<0x20>

This defect is observed when using the digipeating capability built in to the Kantronics KPC-3+ TNC. So you are probably thinking: use it as a KISS TNC and provide digipeating with an external application. See next item.

4.20 Digipeater Causes Long Delay of Packets

It was observed that the position reports, from some moving vehicles, were jumping ahead and back along the route. This was caused by packets being delayed, sometimes for hours, and arriving out of order.

Older KPC-3+ TNCs in KISS mode with an application for digipeater or IGate can introduce very long delays. This has been known for a long time. https://blog.aprs.fi/2011/03/kantronics-kpc3-considered-harmful.html . The workaround might be as simple as connecting two of the RS-232 control lines together.

Here is an episode where we wasted a lot of time finding that it was just another case of this bug: https://groups.io/g/direwolf/topic/buggy_sdr_igates/105889286

It was reported that it was probably fixed in ROM version 9.1 around 2022.

4.21 Incorrect Use of RFONLY or NOGATE

Sometimes you might want to keep a packet only on RF and prevent it from going to APRS-IS. RF to APRS-IS IGate stations should not forward a packet with RFONLY or NOGATE in the digipeater via path.

It should be obvious that you would want to put it at the end of the digi via path.

The following example is wrong. RFONLY was put in the destination field which should have the application identifier for the IGate station.

> NE1CU-10>RFONLY,EKONCT,N3LLO-3,WIDE2*:}N3XKU-7>APMI04,TCPIP,NE1CU-10*:@071128z4010.24N/07450.70WIPHG2230 Fairless Hills, PA; I-gate; 12.9v

From only this, we don't know the software being used by the IGate so we don't easily know what application developer to contact with a defect report.

A web search for RFONLY in the destination field revealed this was a known problem with dxlAPRS. Yep. It's right there in src/udpgate4.c.

4.22 OMG!!! Where do I start with this one?

The first thing we notice is that device identifier is being copied from the third-party payload. One time it is TQ0V4V and other time it is APFIIO. This is wrong; the IGate software should be identifying itself here.

> WA2GUG-15>TQ0V4V,TCPIP,WA2GUG-15,K1EQX-7,N3LLO-3,WIDE2*,RFONLY,NOGATE:}KB1CRN-14>TQ0V4V,WIDE1-1,WIDE2-1,WB2ZII-13,TCPIP,WA2GUG-15*:`e4Tp,Pu/`"4/}Keep on truckin`_1<0x20>
>
> WA2GUG-15>APFII0,TCPIP,WA2GUG-15,K1EQX-7,N3LLO-3,WIDE2*,RFONLY,NOGATE:}N2YTF-3>APFII0,APRSFI,TCPIP,WA2GUG-15*:@011437h4101.46N/07404.18W>359/056/A=000299call me on 146.52!w%C!

TCPIP should appear only the third-party payload, not in the RF via path.

> WA2GUG-15>TQ0V4V,TCPIP,WA2GUG-15,K1EQX-7,N3LLO-3,WIDE2*,RFONLY,NOGATE:}KB1CRN-14>TQ0V4V,WIDE1-1,WIDE2-1,WB2ZII-13,TCPIP,WA2GUG-15*:`e4Tp,Pu/`"4/}Keep on truckin`_1<0x20>

A digipeater should never retransmit a packet that it originally transmitted.

WA2GUG-15>TQ0V4V,TCPIP,WA2GUG-15,K1EQX-7,N3LLO-3,WIDE2*,RFONLY,NOGATE:}KB1CRN-14>TQ0V4V,WIDE1-1,WIDE2-1,WB2ZII-13,TCPIP,WA2GUG-15*:`e4Tp,Pu/`"4/}Keep on truckin`_1<0x20>

Next, we see RFONLY and NOGATE at the end of the RF digipeater via path. Why? No one else does that. It's not in the IGate specification. It was probably a well-meaning but ignorant attempt at preventing this from being sent to APRS-IS by another IGate. Looping is prevented by TCPIP in the encapsulated third party packet.

WA2GUG-15>TQ0V4V,TCPIP,WA2GUG-15,K1EQX-7,N3LLO-3,WIDE2*,RFONLY,NOGATE:}KB1CRN-14>TQ0V4V,WIDE1-1,WIDE2-1,WB2ZII-13,TCPIP,WA2GUG-15*:`e4Tp,Pu/`"4/}Keep on truckin`_1<0x20>

The path in the third-party payload has a lot of extra junk. It should be TCPIP, IGate name, and *. No more, no less.

WA2GUG-15>TQ0V4V,TCPIP,WA2GUG-15,K1EQX-7,N3LLO-3,WIDE2*,RFONLY,NOGATE:}KB1CRN-14>TQ0V4V,WIDE1-1,WIDE2-1,WB2ZII-13,TCPIP,WA2GUG-15*:`e4Tp,Pu/`"4/}Keep on truckin`_1<0x20>

Finally, we find a trailing blank at the end. Where did that come from? This will interfere with recognition of the "_1" device identifier because it is not at the end. (I'm not saying that the IGate did this; there are many other links in the chain where it could have happened. Yaesu radios put a carriage return (0x0d) at the end so someone, along the way, must have changed it to a space.)

WA2GUG-15>TQ0V4V,TCPIP,WA2GUG-15,K1EQX-7,N3LLO-3,WIDE2*,RFONLY,NOGATE:}KB1CRN-14>TQ0V4V,WIDE1-1,WIDE2-1,WB2ZII-13,TCPIP,WA2GUG-15*:`e4Tp,Pu/`"4/}Keep on truckin`_1<0x20>

These IGate stations have the same problems:

N1QQA-10>APWLK,TCPIP,N1QQA-10,KA1GJU-3,WIDE2*,RFONLY,NOGATE:}WLNK-1>APWLK,TCPIP,N1QQA-10*::KB1ZGF :ackKC}

VE2PCQ-3>APSMS1,TCPIP,VE2PCQ-3,WA1PLE-4*,RFONLY,NOGATE:}SMSGTE>APSMS1,VE3OTB-12,TCPIP,VE2PCQ-3*::VA2JW-9 :rej01

4.23 IGate is Trashing UTF-8 Characters

Digipeaters and IGate stations should leave the information part intact. (One little exception: RF to IS IGate removes any trailing CR/LF from the packet because this is the record separator when forwarding to APRS-IS)

I was doing some testing with UTF-8 characters to see how well they were supported on different applications.

Here are some raw packets from aprs.fi, with the timestamps removed to reduce the clutter. These, and most others, preserve the non-ASCII characters. In this context, non-ASCII characters are displayed in hexadecimal.

> **WB2OSZ-5**>APDW18,qAO,KB1TSO:!4237.14NS07120.83W#PHG7140
> Did you know that APRS comments and messages can contain UTF-8 characters?
> <0xce><0xa1><0xce><0xb1><0xce><0xb4><0xce><0xb9><0xce><0xbf><0xce><0xb5><0xcf><0x81><0xce><0xb1><0xcf><0x83><0xce><0xb9><0xcf><0x84><0xce><0xb5><0xcf><0x87><0xce><0xbd><0xce><0xb9><0xcf><0x83><0xce><0xbc><0xcf><0x8c><0xcf><0x82>
>
> **WB2OSZ-5**>APDW18,qAR,W1XM:!4237.14NS07120.83W#PHG7140
> Did you know that APRS comments and messages can contain UTF-8 characters?
> <0xe3><0x82><0xa2><0xe3><0x83><0x9e><0xe3><0x83><0x81><0xe3><0x83><0xa5><0xe3><0x82><0xa2><0xe7><0x84><0xa1><0xe7><0xb7><0x9a>

These changed the non-ASCII characters to question marks.

> **WB2OSZ-5**>APDW18,KB1TSO*,W1MRA,W1MV-1,WIDE2,qAR,NJ1Q:!4237.14NS07120.83W#PHG7140
> Did you know that APRS comments and messages can contain UTF-8 characters?
> ??
>
> **WB2OSZ-5**>APDW18,EKONCT*,WIDE2,qAR,KC1RGS-4:!4237.14NS07120.83W#PHG7140
> Did you know that APRS comments and messages can contain UTF-8 characters?
> ??
>
> **WB2OSZ-5**>APDW18,N3LLO-3,WIDE1,KB1AEV-15*,WIDE2,qAR,W1AW:!4237.14NS07120.83W#PHG7140
> Did you know that APRS comments and messages can contain UTF-8 characters?
> ??

NJ1Q , KC1RGS-4 , and W1AW are all using the same IGate application. There is no commonality in the digipeater paths so that is good evidence that the IGate is the problem. I verified the bug with my own instance of the application and notified the developer.

4.24 Digipeater Corruption of the End of Packet

Here we see a packet which is different depending on which digipeaters it has gone through. One has a trailing carriage return character and the other one doesn't. Someone along the way has modified the information part of the packet. Either some digipeater added a CR at the end or removed it. Either would be wrong. A digipeater should only change the digipeater via path. Modifying the information part will thwart digipeater duplicate suppression.

KB1CRN-14>TRTT2S,N3LLO-3,WIDE1,W1MRA*,WIDE2:`c6<0x1c>l#Ou/`"4R}147.045MHz C100 +060 Keep on truckin'_4<0x0d>

KB1CRN-14>TRTT2S,N1SFT,WIDE1,AB1OC-10,WIDE2*:`c6<0x1c>l#Ou/`"4R}147.045MHz C100 +060 Keep on truckin'_4

More research is needed to narrow this down.

4.25 Defective IGate Implementation or Improper Configuration

The default behavior, of APRS-IS, is to send the following to an IGate station:

(1) "Messages" addressed to stations recently heard by the IGate over RF.
(2) The next position report from the "message" sender if the message was forwarded.
(3) Telemetry metadata and telemetry data from local stations.
(4) Other seemingly random packets for no apparent reason.

An IGate station can optionally ask for more, but not less.

A properly implemented IGate should pass only the first two categories to RF unless the user explicitly enables other packets based on various types of filtering.

On an ordinary day (not APRS Thursday), this IGate is forwarding 339 packets per hour to RF. Average of once each 10.6 seconds.

WZ0C-4>APN20H,WA1PLE-4,WIDE1*,WIDE2-1:}N1EDF-15>APDR16,TCPIP,WZ0C-4*:=4212.14N/07111.22W$007/045/A=-00031 https://aprsdroid.org/<0x0d>

WZ0C-4>APN20H,N3LLO-3,WIDE1*,WIDE2-1:}N1RHY>APOSE,TCPIP,WZ0C-4*:@181932z4236.62N/07118.62W>062/003/A=000132JAMIE 51A TEWKSBURY<0x0d>

They are generally packets where I can hear the source station over RF. This is just redundant useless junk clogging the channel. I don't see anyone else using this particular type of IGate. Here is another example where I'm experimenting with a new HT:

> WB2OSZ-6>APN000:!4237.13N/07120.84Wp000/000
>
> WZ0C-4>APN20H,W1MRA*,WIDE2-1:}WB2OSZ-6>APN000,TCPIP,WZ0C-4*:!4237.13N/07120.84Wp000/000<0x0d>
>
> WZ0C-4>APN20H,W1MRA,WB2OSZ-5*:}WB2OSZ-6>APN000,TCPIP,WZ0C-4*:!4237.13N/07120.84Wp000/000<0x0d>
>
> WZ0C-4>APN20H,W1MRA,AB1OC-10,WIDE2*:}WB2OSZ-6>APN000,TCPIP,WZ0C-4*:!4237.13N/07120.84Wp000/000<0x0d>
>
> WZ0C-4>APN20H,WA1PLE-4,WIDE1*,WIDE2-1:}WB2OSZ-6>APN000,TCPIP,WZ0C-4*:!4237.13N/07120.84Wp000/000<0x0d>

I send a single packet with an empty digipeater path. Four more unwanted copies are QRMing the airwaves. **A carriage return (0x0d) is mysteriously appended.**

Is this a defective IGate implementation or is it bad configuration due to user error?

4.26 IGate Adds Garbage in Front of Received Packet

This was reported in the APRS forum. An IGate station is adding garbage at the beginning of received packets.

> 2025-07-28 16:51:17 EDT: **KA1GJU-8**>TR5R6S,qAR,N2UGS-4: **<<UI>>:**`jJul\ek/`"5p}146.520MHz Kriss - New Hampshire Bound!_1 **[Unsupported packet format]**
>
> 2025-07-28 22:32:34 EDT: **KD2YTC-10**>TRUV0Z,K2ILH-2*,WIDE2-1,qAR,N2UGS-4: **<<UI>>:**`jJSl")YY`"6@}_3 **[Unsupported packet format]**

Notice how it is inserting "<<UI>>:".

Many of the packets generated are invalid.

> 2025-08-04 09:51:08 EDT: **N2UGS-4**>APWW11,TCPIP*,qAC,T2CSNGRAD::@135106h4258.40N/07845.96W#/A=000590PHG5160 One-

Hop WNY Digi 444.0000 **[Invalid message packet]**

5 decode_aprs application

The direwolf software TNC includes a utility that will interpret the contents of an APRS packet and point out many types of errors.

Examples:

(1) Is this packet correct?

N1NW>T1ST8T,EKONCT,W1MRA,N3LLO-3,WIDE2*:'d^9l <0x1c>#/]N1NW 146.730 TONE 156.7<0x0d>

On Linux, you can pipe a packet into stdin like this:

$ echo "N1NW>T1ST8T,EKONCT,W1MRA,N3LLO-3,WIDE2*:'d^9l <0x1c>#/]N1NW 146.730 TONE 156.7" | decode_aprs

N1NW>T1ST8T,EKONCT,W1MRA,N3LLO-3,WIDE2*:'d^9l <0x1c>#/]N1NW 146.730 TONE 156.7
"146.730" in comment looks like a frequency in non-standard format.
For most systems to recognize it, use exactly this form "146.730MHz" at beginning of comment.
"156.7" in comment looks like it might be a CTCSS tone in non-standard format.
For most systems to recognize it, use exactly this form "T156" at near beginning of comment, after any frequency.
MIC-E, Generic digipeater, Kenwood TM-D700, In Service
N 41 34.8400, W 072 06.2900, 0 km/h (0 MPH), 146.730 MHz, PL 156.7
N1NW 146.730 TONE 156.7

The original packet is echoed in green.
Error messages appear in red.
An interpretation is in blue.
The final line is the comment after removing any embedded information.

(2) What does this mean?

N83MZ>T2TQ5U,WA1PLE-4*:`c.l+@&'/'"G:} KJ6TMS|!:&0'p|!w#f!|3

Trying to quote this would be challenging because it contains both " and ' characters. You could either:

- Run decode_aprs with no command line options and then type the APRS packet. Or
- Put the packet(s) into a file and specify the file name on the command line.

$ decode_aprs test.txt

N83MZ>T2TQ5U,WA1PLE-4*:`c.l+@&'/'"G:} KJ6TMS|!:&0'p**|!w#f!**|3

MIC-E, Small Aircraft (original primary symbol), Byonics TinyTrack3, In Service
N 42 41.5502, W 071 18.8076, 283 km/h (176 MPH), course 210, alt 1764 m (5787 ft)
Seq=25, A1=470, A2=625
KJ6TMS

In this case, parts of the comment field are interpreted as embedded data.

|!:&0'p|	Is Base-91 telemetry data.
!w#f!	Added more digits of resolution to the location.
|3	is the sending device type identifier.

This leaves only “KJ6TMS” for the comment text. The source “N83MZ” looks like a tactical callsign (aircraft tail number in this case) so the comment has the official radio callsign to keep things legal.