9 Shenanigans with APRS and AX.25 for Covert Communications

by Vogelfrei

This little document details some shenanigans involving APRS and its underlying AX.25 protocol, including but not limited to covert channels, steganography, avoiding detection by normal users and leveraging Internet infrastructure for worldwide covert communication.

Covert channels in radio packet protocols have been investigated in the past.[21] Although the regulations for amateur radio operation explicitly forbid hiding, encoding, or encrypting communications in any form, it is nonetheless a challenging and fruitful field for experimentation.

I had been researching the topic for a while, and informally mentioned this to my neighbors Travis and Muur, who—it turned out—had been working on PSK31. They requested an article to follow theirs, PoC‖GTFO 8:4. So enjoy this short piece, and look out for more elaborate tricks and tools for all your booklegging communication needs, because the world is almost through![22]

The APRS protocol (Automatic Position Reporting System), originally developed by Bob Bruninga (WB4APR), has its roots in the necessity to track the position and telemetry data of vehicles, weather stations, and hikers.

APRS is built on the AX.25 protocol, an amateur variant of the commercial X.25 protocol you'll fondly remember from Phrack 45:8. Despite the amateur nature of its deployment, there is an impressively large infrastructure of Internet gateways, digipeaters, weather stations, and other kinds of nodes. The International Space Station (ISS) itself has an APRS-capable digipeater on-board, and radio operators across the globe engage in packet radio messaging through the station and other satellites.

Perhaps the most interesting feature of APRS, besides the fact that it supports exchanging all kinds of information, is the way the data is routed between uncoordinated nodes over large areas. It is this decentralized, connection-less nature that makes APRS ideal for covert communication purposes.

9.0.1 Frequencies and Equipment

Now that you have a general idea of what APRS is and what it might be useful for, you should know which frequencies are designated for APRS transmissions. Frequencies vary by country, but as a general rule, North America uses 144.390 MHz while Europe and Africa use 144.800 MHz.

For testing and experimentation purposes, start with a cheap hand-held radio such as the Baofeng UV5R from China. It is capable of transmitting in the 2m and 70cm bands, and can easily be connected to your computer's sound card. This will allow you to immediately test software modems and get your feet wet with APRS and other packet radio protocols.

If you would like to get fancy, I recommend two additional pieces of equipment. Get a dual-band radio with TNC support, such as the Kenwood TM-D7xx or TH-D72A. The TNC will interpret packets in hardware, freeing you from DSP headaches. You will also want a general purpose wide-band receiver with discriminator (unadulterated audio) output; ordinary folks call this a scanner.

9.1 The Protocol

As mentioned before, APRS uses AX.25 for transport. More specifically, APRS data is contained in AX.25 Unnumbered Information (UI) frames, in the information field. The protocol is completely connectionless; there is neither state nor any expectation of a response for a given packet.[23] This is rather handy for simple systems, since you will only need a single packet consumer, and the rest of your state machine is entirely up to you. Because of its simplicity, APRS can be easily implemented in microcontrollers.

A simple APRS message packet looks as follows:

[21] `jt64stego` by Drapeau (KA1OVM) and Dukes, 2014

[22] So says the preacher man but... I don't go by what he says.

[23] This is the exact opposite of your Wi-Fi, where every data frame is acknowledged, and no more data is sent unless either the ACK arrives or a timeout is reached.

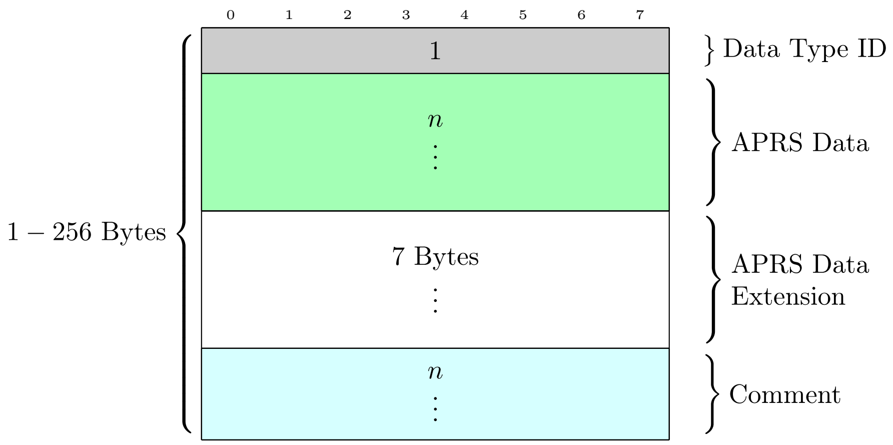

Figure 6: APRS Data contained in the AX.25 information field

```
NOCALL-9>N1CALL-9,WIDE1-1,WIDE2-2::N1CALL-9 :This is a test for APRS messages{1
```

Dissecting its structure, we will find:

1. The path element: `NOCALL-9>N1CALL-9,WIDE1-1,WIDE2-2`
2. A colon (:) delimiting the end of the path and the beginning of the packet data.
3. The packet type identified by a single character, `:` for messages.
4. After that, whatever format the packet type specifies. In the case of a message, a colon-delimited recipient callsign, followed by the text and a `{` bracket followed by a number, indicating the line of the message, starting at one.

The comment field is also susceptible to abuse, limited to printable ASCII data as the specification demands, “The comment may contain any printable ASCII characters (except `|` and `~`, which are reserved for TNC channel switching).” Depending on the DTI, the Comment field is used to include additional information besides what is sent in the Data field, mostly for telemetry uses. Coordinates are encoded using Base-91.

The wealth of information provided in the original protocol specification should be more than enough to figure out ways to conceal your own data in different packet types. Of particular interest are the mechanisms for compressed coordinates and telemetry, weather reports, and bulletin messages. While these have size limitations, leveraging the unused DTIs as described in the next section allows for crafty ways to chain multiple packets together.

9.2 Abusing Unused Data Type Identifiers (DTI)

The APRS protocol defines multiple DTIs as unused or forbidden. These are often ignored by software and TNCs in actual radios, making them an ideal target for creative reuse. Because it would be trivial to detect and actively monitor for intentional use of the unused DTIs, a better approach is to leverage them in a way that provides somewhat plausible deniability.

1. Prepare APRS Data contents for a given DTI.
2. Find nearest unused DTI, possibly identifying the unused DTIs that require the least amount of bits to corrupt so that the DTI isn’t “too far” from the one corresponding to the data we have prepared.

ID (char)	Data Type	Valid DTI neighboring?
0x22	Unused	0x21 (position without timestamp or WX) and 0x23 (WX)
0x26	Reserved ("map feature")	0x25 (MicroFinder) and 0x27 (Mic-E or TM-D700 data)
0x28	Unused	0x27 and 0x29 (Item)
0x41-0x53	Unused	Only adjacent (0x40 and 0x54)
0x2c	Experimental/Unused	(none)
0x2e	Reserved (Space weather)	0x2f (position with timestamp sans messaging)
0x30-0x39	Do not use	0x3a (Message)

Table 1: Some of the unused Data Type Identifiers in the APRS protocol

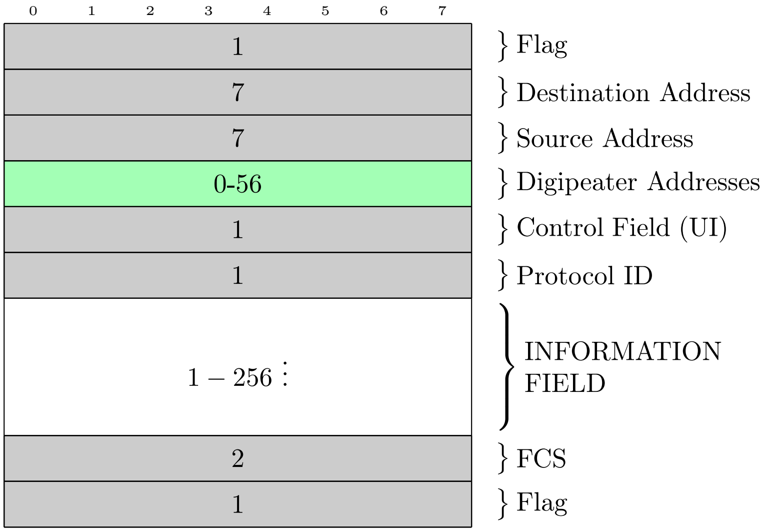

Figure 7: AX.25 Unnumbered Information (UI) frame structure

3. Proceed to send the packet contained an invalid DTI that is unused yet contains seemingly valid data for an adjacent DTI.

Unused DTIs that are one position away from another include 0x21 and 0x22 (position without timestamp versus unused) Table 1 contains some of the interesting unused identifiers up for grabs; please refer to the APRS Protocol Reference[24] for the rest of them. DTIs involved in TNC operation should be avoided, unless the TNC behavior can be abused constructively.

The benefit of hiding data in an otherwise valid APRS Data segment with an incorrect (unused) DTI is that clients—including built-in TNCs—will ignore the packet and not attempt to decode its contents.

9.2.1 Third-party and User Defined Packets

Two special DTIs exist that allow for packet-in-packet protocol tricks: the third-party and user-defined packets. These have special quirks associated with them, and the way TNCs handle them is not standardized. This is both a good and a bad thing. For instance, the Kenwood TM-D7xx's built-in TNC will ignore third-party packets entirely if it cannot parse them.

[24] `unzip pocorgtfo09.pdf aprs101.pdf`

However, Internet Gateways will also ignore all user-defined packets and impose additional restrictions the third-party DTI. This is the biggest motivator for actually reading the source code of APRS Internet gateway software. For example:

```
static int parse_aprs_body(struct pbuf_t *pb, const char *info_start)
{
...
  case '{':
    pb->packettype |= T_USERDEF;
    return 0;

  case '}':
    pb->packettype |= T_3RDPARTY;
    return parse_aprs_3rdparty(pb, info_start);
```

```
NOCALL-9>N1CALL-9,WIDE1-1,WIDE2-2::N1CALL-9 :This is a test for APRS messages{1
```

9.3 Internet Gateways

Gateways between the Internet and APRS radios are known as Internet Gateways or iGates. Typically iGates are used to forward APRS beacons heard over radio to some website, but there are a lot more interesting things we could do with them.

9.3.1 Tricks with iGates

Some iGates support transmitting data from the Internet out to radio, effectively bridging the local RF spectrum to the APRS-IS network.

There is no official way to list iGates, so our best bet is connecting to the backbone servers they report to, passively listening for frames and beacons that announce their presence. We would also like to distinguish iGates that are capable of transmitting from those that only receive. When we find some such iGates, they allow us to perform some gnarly tricks!

We can send an APRS message from an Internet-only host in Asia to an individual driving in Pittsburgh with only a radio receiver and a TNC. Hide locations of control sites by first proxying your packets through the Internet iGates, only to target your local RF nodes through a separate, sacrificial iGate bridge.

The system is only limited by APRS-IS rules in terms of traffic congestion control. Because all RF nodes receive from and transmit to the same frequency, overlapping transmissions can and will reduce the ratio of successfully decoded packets for everyone else. Therefore, be neighborly!

Traffic caps are enforced by the iGate operator's configuration. Commonly a given node, as identified by its callsign and SSID, will only be able to use the Internet-RF bridge for transmitting a fixed number of packets each minute. This is to prevent accidental jamming of the RF channel.

9.3.2 Packet Validation and RF Digipeating

Some architectural limitations of APRS need to be considered carefully. First, most iGates in the APRS-IS network will only digipeat packets to the RF side if the station is located within a fixed radius of so many kilometers. Second, we might not get to know if a given area has an iGate capable of bridging RF, or transmitting to RF. We can't simple wait for a response, as APRS is a response-less protocol. Third, packets marked `RFONLY` in their path won't reach APRS-IS. Packets marked TCPIP won't reach RF nodes. iGates forcing or restricting either will be dead-ends if we aim to bridge over APRS-IS. Finally, user-defined packets are ignored by most of the APRS-IS infrastructure. For example, `aprsc` ignores them. Third-party packets are allowed, with caveats.

9.3.3 Bypassing Validation

There are a few ways to bypass the restrictions imposed on bridging RF in iGates that require geographical proximity.

You can try to spoof your location by sending a beacon positioned at fake coordinates near the iGate. You can then send your actual data packets, remembering to regularly send a position beacon to the iGate to remain in the last-heard list.

You could limit use of user-defined packets to RF side, operating a a rogue iGate that does *not* ignore them, instead transforming them to third-party or steganographic standard packets, delivered to APRS-IS. User-defined packets are not displayed by most equipment. This also applies to unused or obscure DTIs.

To avoid potential roadblocks, the following considerations may help. If trying to reach the RF side, do not use (and verify that the iGate/APRS-IS nodes don't use) `TCPIP` in the path. If trying to reach the Internet side, do not use `RFONLY` in the path. To avoid packet drops from rate limiting, throttle your packets, sending one every one to five minutes.

Albeit completely illegal on the actual air, as an experiment in a controlled environment, automatically generated callsigns can be rotated to avoid being detected or banned from the system.[25] Finally, client version strings, as used during registration with APRS-IS nodes, could be rotated and mimic real clients.

Looking up standard TCP/IP "pivoting" techniques may help for accessing the APRS-IS network, but first and foremost, remember to be neighborly.

9.3.4 International Space Station (ISS) and APRS

Space, the final frontier! It suffices to say that a digipeater installed onboard the ISS makes APRS into the tool of choice for legal ruckus communications on a worldwide scale. So as long as the TNC of the ISS' radio validates your packets, you can deliver your covert messages in a fully decentralized fashion![26]

Whether commercial TNCs out there relay packets with unused DTIs is a question left to the reader as an exercise.

9.4 Parting words: legal status of subterfuge in radio communications

Amateur radio laws generally prohibit steganography and also encryption, with a few narrow exceptions.[27] For example, the US Electronic Code of Federal Regulations §97.309 states, *RTTY and data emissions using unspecified digital codes* **must not be transmitted for the purpose of obscuring the meaning of any communication**.[28][29] Governments do monitor the airwaves where they care about them the most, and having your antennas, expensive equipment, or house ransacked sucks. Also keep in mind that amateur radio is self-policing; if you mess up and create a nuisance that affects everyone else, your future experiences with that small, tight-knit, but global community may be seriously soured. So be neighborly, have fun, and stay safe!

—Vogelfrei

[25] Don't do this. Acting like an asshole on the radio is the surest way to convince a brilliant RF engineer to spend his retirement hunting you down.

[26] *In Heinlein's "Between the planets", 1951, the same celestial path of the Circum-Terra station is used for a much less benign purpose: worldwide delivery of nukes. That book also introduced the idea of stealth technology vehicle with a radar-reflecting surface, before any scientific publications on the subject. Welcome to classic 1950s Sci-Fi.—PML*

[27] `unzip pocorgtfo09.pdf encham.html #Encryption and Amateur Radio by KD0LIX`

[28] `unzip pocorgtfo09.pdf part97.pdf`

[29] Also note §97.217: *Telemetry transmitted by an amateur station on or within 50 km of the Earth's surface is not considered to be codes or ciphers intended to obscure the meaning of communications.*