

# APRX Software Requirement Specification

## Table of Contents

1	APRX Software Requirement Specification.....	2
1.1	Purpose:.....	2
1.2	Usage Environments:.....	3
2	Treatment rules:.....	4
2.1	Basic IGate rules:.....	4
2.2	Low-Level Transmission Rules:.....	5
2.3	Low-Level Receiving Rules:.....	6
2.4	Digipeater Rules:.....	6
2.5	Additional Tx-IGate rules:.....	7

## 5 **1 APRX Software Requirement Specification**

6 This is *Requirement Specification* for a software serving in Amateur Radio APRS service.

7 Reader is assumed to be proficient with used terminology, and they are not usually  
8 explained here.

### 9 **1.1 Purpose:**

10 This describes algorithmic, IO-, and environmental requirements for a software doing any  
11 combination of following three tasks related to APRS service:

- 12 1. Listen on messages with a radio, and pass them to APRSIS network service
- 13 2. Listen on messages with a radio, and selectively re-send them on radio
- 14 3. Listen on messages with a radio, and selectively re-send them on radios on other  
15 frequencies
- 16 4. Receive messages from APRSIS network, and after selective filtering, send some of  
17 them on radio

18

19 Existing *aprx* software implements Receive-Only (Rx) IGate functionality, and the purpose  
20 of this paper is to map new things that it will need for extending functionality further.

21

22

23

## 24 **1.2 Usage Environments:**

25 The *aprx* software can be used in several kinds of environments to handle multiple tasks  
26 associated with local APRS network infrastructure tasks.

27 On following one should remember that amateur radio transmitters need a licensed  
28 owner/operator, but receivers do not:

- 29 1. License-free Receive-Only (RX) IGate, to add more “ears” to hear packets, and to  
30 pipe them to APRSIS
- 31 2. Licensed bidirectional IGate, selectively passing messages from radio channels to  
32 APRSIS, and from APRSIS to radio channels, but not repeating packets heard on a  
33 radio channel back to a radio channel.
- 34 3. Licensed bidirectional IGate plus selectively re-sending of packets heard on radio  
35 channels back to radio channels
- 36 4. Licensed system for selectively re-sending of packets heard on radio channels back  
37 to other radio channels, and this without bidirectional IGate service.
- 38 5. Licensed system for selectively re-sending of packets heard on radio channels back  
39 to radio channels, and doing with with “receive only” IGate, so passing information  
40 heard on radio channel to APRSIS, and not the other way at all.

41

42 In more common case, there is single radio and single TNC attached to digipeating (re-  
43 sending), in more challenging cases there are multiple receivers all around, and very few  
44 transmitters. Truly challenging systems operate on multiple radio channels. As single-  
45 TNC and single-radio systems are just simple special cases of these complex systems,  
46 and for the purpose of this software requirements we consider the complex ones:

- 47 1. 3 different frequencies in use, traffic is being relayed in between them, and the  
48 APRSIS network.
- 49 2. On each frequency there are multiple receivers, and one well placed transmitter.
- 50 3. Relaying from one frequency to other frequency may end up having different rules,  
51 than when re-sending on same frequency: Incoming packet retains traced paths,  
52 and gets WIDEN-N/TRACEN-N requests replaced with whatever sysop wants.

53

54

## 55 2 Treatment rules:

56 Generally: All receivers report what they hear straight to APRSIS, after small amount of  
57 filtering of junk messages, and things which explicitly state that they should not be sent to  
58 APRSIS.

### 59 2.1 Basic IGate rules:

60 General rules for these receiving filters are described here:

61 <http://www.aprs-is.net/IGateDetails.aspx>

62  
63 Gate all packets heard on RF to the Internet (Rx-IGate) EXCEPT

- 64 1. 3rd-party packets (data type 'J' ).  
65 3rd-party packets should have all before and including the data type stripped  
66 and then the packet should be processed again starting with step 1 again.  
67 2. generic queries (data type '?' ).  
68 3. packets with TCPIP, TCPXX, NOGATE, or RFONLY in the header (last 2 are  
69 optional).

70  
71 Gate message packets and associated posits to RF (Tx-IGate) if

- 72 1. the receiving station has been heard within range within a predefined time  
73 period (range defined as digi hops, distance, or both).  
74 2. the sending station has not been heard via RF within a predefined time  
75 period (packets gated from the Internet by other stations are excluded from  
76 this test).  
77 3. the sending station does not have TCPXX, NOGATE, or RFONLY in the  
78 header.  
79 4. the receiving station has not been heard via the Internet within a predefined  
80 time period.

81 A station is said to be heard via the Internet if packets from the station  
82 contain TCPIP\* or TCPXX\* in the header or if gated (3rd-party) packets are  
83 seen on RF gated by the station and containing TCPIP or TCPXX in the 3rd-  
84 party header (in other words, the station is seen on RF as being an IGate).

85 Gate all packets to RF based on criteria set by the sysop (such as call sign, object  
86 name, etc.).

87  
88 With more advanced looking inside frames to be relayed, both the digipeater and Tx-IGate  
89 can use filtering rules, like “packet reports a position that is within my service area.”

90

91

92 From multiple receivers + single (or fewer) transmitter(s) follows, than when a more usual  
93 system does not hear what it sent out itself, this one will hear, and its receivers must have  
94 a way to ignore a frame it sent out itself a moment ago.

95 Without explicit “ignore what I just sent” filtering, an APRS packet will get reported twice to  
96 APRSIS:

97 rx -> igate-to-aprsis + digi -> tx -> rx -> igate-to-aprsis + digi (dupe stops)

98 Digipeating will use common packet duplication testing to sent similar frame out only once  
99 per given time interval (normally 30 seconds.)

100

## 101 **2.2 Low-Level Transmission Rules:**

102 These rules control repeated transmissions of data that was sent a moment ago, and other  
103 basic transmitter control issues, like persistence. In particular the persistence is fine  
104 example of how to efficiently use radio channel, by sending multiple small frames in quick  
105 succession with same preamble and then be silent for longer time.

- 106 1. Normal digipeater duplicate packet detection compares message source,  
107 destination, and payload data against another expiring storage called “duplicate  
108 detector”. Digipeater and Tx-IGate will ignore packets finding a hit in this storage.
- 109 2. A candidate packet is then subjected to a number of filters, and if approved for it,  
110 the packet will be put on duplicate packet detection database (one for each  
111 transmitter.) See Digipeater Rules, below.
- 112 3. Because the system will hear the packets it sends out itself, there must be a global  
113 expiring storage for recently sent packets, which the receivers can then compare  
114 against. (Around 100 packets of 80-120 bytes each.) This storage gets a full copy  
115 of packet being sent out – a full AX.25 frame.

116 Also, transmitters should be kept in limited leash: Transmission queue is less than N  
117 seconds ( < 5 ? ), which needs some smart scheduling coding, when link from computer to  
118 TNC is considerably faster.

119 Original KISS interface is defined as “best effort”: if TNC is busy while host sends a frame,  
120 the frame will be discarded, and “upper layers” will resend. In APRS Digipeating, the  
121 upper layer sends the packet once, and then declares circa 30 second moratorium on  
122 packets with same payload.

123

124

### 125 **2.3 Low-Level Receiving Rules:**

- 126 1. Received AX.25 packet is compared against “my freshly sent packets” storage,  
127 matched ones are dropped. (Few transmitters, multiple receivers hear them.)
  - 128 2. Received packet is validated against AX.25 basic structure, invalid ones are  
129 dropped.
  - 130 3. Received packet is validated against Rx-IGate rules, forbidden ones are dropped  
131 (like when a VIA-field contains invalid data.)
  - 132 4. Packet may be rejected for Rx-IGate, but it may still be valid for digipeating!  
133 For example a 3<sup>rd</sup> party frame is OK to digipeat, but not to Rx-IGate to APRSIS!
- 134 Divide packet rejection filters to common, and destination specific ones.

135

### 136 **2.4 Digipeater Rules:**

137 Digipeater will do following for each transmitter:

- 138 1. Compare candidate packet against duplicate filter, if found, then drop it. (Low-level  
139 transmission rules, number 1)
- 140 2. Count number of hops the message has so far done, and...
- 141 3. Figure out the number of hops the message has been requested to do  
142 (e.g. “OH2XYZ-1>APRS,OH2RDU\*,OH2RDK\*,WIDE7-5: ...” will report that there  
143 was request of 7 hops, so far 2 have been executed.)
- 144 4. If either of previous ones are over any of configured limits, the packet is dropped.
- 145 5. FIXME: WIDEn-N/TRACEn-N treatment rules  
146 Possible additional keywords for cross-band digipeating?
- 147 6. Multiple filters look inside the message, and can enforce a rule of “repeat only  
148 packets within my service area.”
- 149 7. FIXME: Cross frequency digipeating? Treat much like Tx-IGate?  
150  
151 Relaying from one frequency to other frequency may end up having different rules,  
152 than when re-sending on same frequency: Incoming packet retains traced paths,  
153 and gets WIDEn-N/TRACEn-N requests replaced with whatever sysop wants.
- 154 8. Cross band relaying may need to add both an indication of “received on 2m”, and  
155 transmitter identifier: “sent on 6m”:  
156 “OH2XYZ-1>APRS,RX2M\*,OH2RDK-6\*,WIDE3-2: ...”  
157  
158 This “source indication” may not have anything to do with real receiver identifier,  
159 which is always shown on packets passed to APRSIS.

160

161

162 **2.5 Additional Tx-IGate rules:**

163 The Tx-IGate can have additional rules for control:

- 164 1. Multiple filters look inside the message, and can enforce a rule of “repeat only  
165 packets within my service area,” or to “limit passing message responses only to  
166 destinations within my service area”
- 167 2. Basic gate filtering rules:
- 168 1. the receiving station has been heard within range within a predefined time  
169 period (range defined as digi hops, distance, or both).
- 170 2. the sending station has not been heard via RF within a predefined time period  
171 (packets gated from the Internet by other stations are excluded from this test).
- 172 3. the sending station does not have TCPXX, NOGATE, or RFONLY in the header.
- 173 4. the receiving station has not been heard via the Internet within a predefined time  
174 period.
- 175 A station is said to be heard via the Internet if packets from the station contain  
176 TCPIP\* or TCPXX\* in the header or if gated (3rd-party) packets are seen on RF  
177 gated by the station and containing TCPIP or TCPXX in the 3rd-party header (in  
178 other words, the station is seen on RF as being an IGate).
- 179 3. More rules ?

180

181