Intelligent Filtering of the APRS Internet Gateway Data Streams

Darryl Smith, B.E., VK2TDS
POox 169
Inglebum NSW 1890
Australia
Phone: +61 412 929 634
Daryl@radio-active.net.au

ABSTRACT: APRS is a fast growing mode throughout the Amateur Radio world, thanks to its combination of computers, radio, packet and the Internet. Such popularity has increased the amount of data on the APRS network to a point where many users, especially those outside North America, are overwhelmed by the volume, or in an attempt to reduce the traffic volume do not enjoy the full benefits iflnternet Connected APRS. This paper describes work by the Author intopltering the data stream intelligently to create manageable local networks.

Introduction

In the past 3 years since Internet gating of APRS packets was started, APRS.NET traffic volumes have increased dramatically. Although I have been unable to find any documentation on past traffic volumes, measurements indicate that a complete stream from APRS.NET in mid-2000 would consume more than 25-50 Mbytes per day.

To gate this volume of traffic to RF would require probably 10 times the transmission rate as provided by the current 1200 bps packet modems given the highly distributed nature of APRS and ALOHA networking.

Even the most basic data stream, consisting of only message and related position reports consumes upwards of 4 Mbytes per day, placing a significant load on any transmitting equipment with little benefit to those receiving the transmissions.

The number of stations world wide is beginning to take its toll on th.e APRS software. We have recently seen instances where the number of stations on the network caused some APRS software to fail. Redraw times have increased so that only the fastest computers can realistically be used to use APRS.

AFilter for instance was written so that end users could filter their own data streams, improving the performance and stability of APRS applications.

Review of Traffic Volumes

Analyzing the APRS.NET traffic stream for about 15 minutes reveals some interesting results. The USA accounts for 79.8% of the packets in this period that I logged the APRS.NET Server. Adding Canada increased the North American trafficto 89.2%.

Country	W	m
USA	786	79.; 0 /c
Canada	92	9.3%
Netherlands	23	2.3%
Australia	16	1.6%
Switzerland	14	1.4%
UK	11	1.1%
Brazil	8	0.8%
Hungary	7	0.7%
Portugal	6	0.6%
New Zealand	5	0.5%
Honduras	5	0.5%
(sermany	5	0.5%
[taly	4	0.4%
South Korea	3	0.3%

Unfortunately, most of the USA based traffic is of little use for the rest of the world. In fact most of the USA traffic is of little use to most other USA based amateurs. By eliminating most of this data stream, the volume of data is far more manageable, and scalable.

In some areas the current trend is to place the entire APRS data stream onto RF. Whilst this is a noble endeavor, it does tend to cause some problems when the data stream contains more information than the channel can support.

Data Segregation

The solution to the data overload problem is to somehow reduce the volume of data that individual users are required to deal with.

Several methods can be used to segregate traffic

- Originating Callsign (Region)
- Position in the world
- Type (Message, Position, etc)
- Redundancy of data

A combination of filters based on these conditions will yield a reduced flow of information, without decreasing the relevant information. The APRSd software already has filters to remove redundant data. Apart from these it has very limited filtering capabilities.

Of course, MICE Emergency Packets should be passed unconditionally.

Hierarchical Networks

Anyone who has researched the history of the Internet will know that in the beginning it was a very flat network, with flat protocols to support the network. Even the naming of machines was done on a single-level structure.

As the number of users grew, the old structures were no longer able to support the increased usage. Therefore the intemet has been transformed into a highly distributed hierarchical structure.

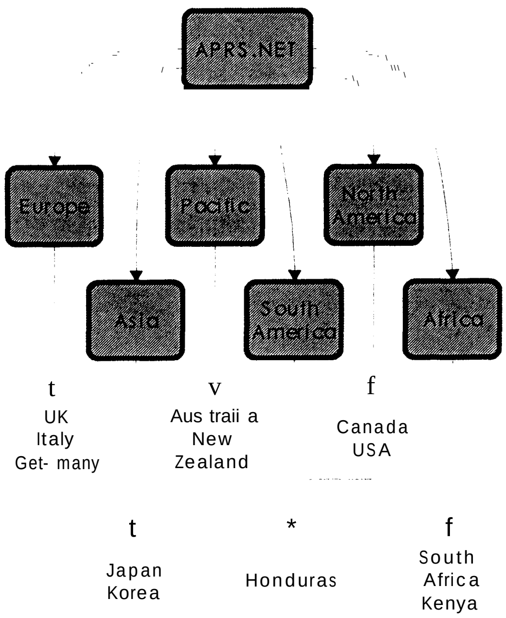
The APRS IGATE system has similarly grown, and is becoming increasingly fragmented and hierarchical. It is growing thanks to the proliferation of Open Source software, as well as easier to use APRS software allowing an IGATE to be setup without actually knowing what you are doing, at almost zero cost.

The haphazard connections between IGATES has caused some problems with APRS messaging. Whilst packets are getting from the extremities of the network to the main APRS server, messages sent to the stations providing these positions will often not get through.

The cause is due to the fact that many of the IGATE systems only feed data into the network, and do not retrieve data because of the sheer volume of data.

Proposed Hierarchy Of Servers

I am proposing that APRS IGATE data be filtered by the region, initially by callsign. Once this happens, filtering by region can be added. Those that prefer the present situation where they can see all the world wide stations will still be able to connect to APRSNET.



In this proposal, there would be a root APRS server, and a number of regional servers. These regional servers would be permanently connected to the root server exchanging data much as happens with IGATES today.

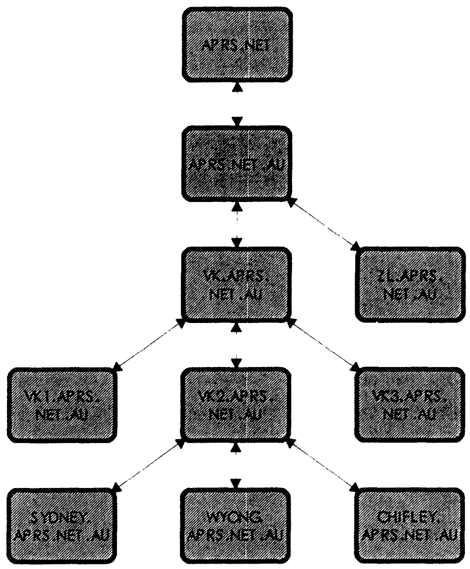
Please note that I have not tried to include any discussions about backup servers in this paper. They are able to operate in a manner similar to the present system with *second. aprs. net.*

Users would be encouraged to connect to a regional server, but there would be no requirement for this. It is anticipated that the APRS.NET root server would become the de-facto USA server under this plan.

As you go further down the chain you would see a structure similar to the one appearing on the next page.

This does not however have any advantage over the current structure of IGATE's except the structure of naming. Something else is required before this will give gains required for the whole IGATE system.

To obtain any improvements some form of filtering is required. between the levels of the network.



Filtering of Data By Callsign

Australia and New Zealand have some particular properties that both help and hinder the development of our national APRS networks

Australia has a very structured callsign system by state. The Number following the VK in Australian callsigns indicate the STATE that the Amateur lives in. It is rare to find an amateur operating from a state other than what his callsign would indicate, given the size of the each state.

New Zealand has a similar structure, although because of it's size there is little use breaking down traffic much further.

Visitors to Australia must obtain a LOCAL callsign before they may legally operate in Australia, which makes the structure even more viable. However much of the rest of the world does not have such stringent requirements, so another solution is required.

Filtering of Data by Position

Because of the structure of the Australian and New Zealand callsigns I have only recently attempted to filter based on positions. This is the logical next step so that users with non-local callsigns are able to automatically use the local IGATE system.

Development of software to filter the data stream by position has just been completed. The filtering technique being developed is based in Richard Parry's perlAPRS software. It takes a list of valid grid squares, and then uses these to validate the position.

Rules - Filtering Philosophy

Child to Parent IGATE

A number of basic rules can be applied to filter data between the IGATE and the parent IGATE. Any packets that do not satisfy one of these rules will be discarded.

1. All packets from VK*
2. All packets from stations inside VK
3. All messages to VK* or stations inside VK
4. All packets from a station who's packets are permitted under points ##3, for 30 minutes after the last permitted packet.

Rule 1: Allows concentration of all VK* data even when an alternate IGATE is used.

Rule 2: Allows any station that does not have a VK callsign to be treated as if they did have a VK callsign.

Rule 3: Any messages to VK stations, or other stations operating inside VK should be allowed in.

Rule 4: Allows positions of any stations sending messages to VK to show up on the server. This rule may need to be changed later so that their messages to non-VK stations do not appear - As only one side of the conversation would be heard anyway. The 30 minute limit allows keeps the link open as long as the users continue to converse.

Child to Parent IGATE

Some IGATE data is not relevant apart from for users of that IGATE. Therefore the following rules are used to filter the data stream. All packets will be passed, except if they match any of these rules.

1. Messages to 'javaMSG' and to 'USERLIST'.
2. Packets to 'FBB'
3. Packets to 'MAIL' where the first characters of the payload are 'BBS'
4. Packets that have malformed contents.

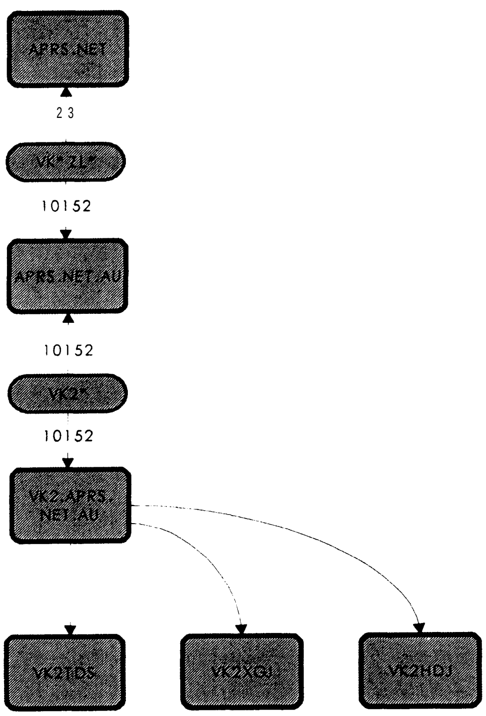
These rules reduce the amount of data being sent to the parent quite significantly.

Users?

Having a look at the: diagram below one important question is 'Where does a user connect?'. The answer is basically wherever they want to. Generally the answer would be to connect to the server offering country-wide data. However if they were wanting to IGATE their information from other IGATE's in their state only, they might want to connect to their State APRS server.

In either case, any packets they receive from RF and place back onto the APRS network would filter back to the root APRS server, as well as down to the State based APRS server.

Messages will be delivered regardless of the where the user is connected, provided the connection point is within the filtering range. Messages will be fed up the chain no matter what filtering rules are in place.



Filtering Software

The filtering software I have written has been coded in Per1 running under Linux. I chose Per1 as it is a very quick prototyping language, with the power of 'C' and the flexibility of Basic. As it is basically an interpreted language, it is very quick to modify code and see how the changes work.

The software consists of three modules. They are

- perlFilter.pl
- perlTelnet.pl
- perlAPRS.pl

perlFilter is the main filtering program. It contains all the rules, and logic deciding what packets are allowed in either direction. It connects to the up-stream and down-stream APRSd servers, providing all the data communications between the two. perlFilter actually authenticates itself with each server so that the connections are bi-directional.

perlTelnet is a specialized TELNET program. Whilst most telnet programs quit when the connection is closed, perlTelnet then attempts a connection to the next server on the list. If no servers can be contacted, it continues until it finds one that can..

A basic structure of the software appears in the diagram following. In this diagram, the boxes with the partial callsigns ("VK*", "ZL*" and "VK2*") are the perlFilter programs, with the filter properties listed.

In this way, APRS.'NET.AU only receives packets from APRSNET that are relevant to it - All VK and ZL packets, as well as packets to those stations.

Likewise VK2.APRS.NET.AU only receives packets from APRS.NET.AU that are relevant to VK2. This arrangement reduces the load on servers and the underlying communications network.

I have also modified Richard Parry's perlAPRS software to check if the location of a mobile station is within the location specified by the configuration file.

perlFilter does NOT filter packets from stations that connect directly into the IGATE. What would be ideal is for the capability for APRSd to use external filters on all incoming data streams automatically.

Operation

The software has now been operational for several weeks, and is working well. At times it causes the operation of APRS to become less intuitive - by requiring users to send messages to stations outside the filter range before they will appear on the screen. Until the user appears on the screen, users do not normally send messages.

In other words, users must register their locations, or an association with the location (by sending a message to that location) before their data will be passed. This geo-referencing should be an integral part of all wide area AVL systems

I found that soon after I started messaging a user in the USA, his position appeared on my map. Also his messages to other users in the USA appeared on the raw data coming from my APRS server.

I found that the reduction in data volume was a great advantage in operating this way. I have found that users are generally happier with the reduced volumes, as they were being overloaded with information.

I am now looking at how to integrate similar software into APRSd as a hookpoint to improve the filtering.

Conclusion

Just as the telegraph, telephone and Internet have grown from flat networks to hierarchical networks, APRS networking must also mature in a similar manner. Unlike those other networks, the basic protocols will not need to change for this shift, just the servers, and their interconnections.

The work I have done on creating hierarchical servers is just another step in the evolution of APRS towards a scalable network oriented system. The software I have developed allows users to add customized filters to the data stream in line with local requirements.

Software Availability

The software is available on the following WWW site:

http//radio-active.net.au

It is written for Redhat LINUX, but should run on almost any platform that supports Per1 and multi-processing. It does not even need to be running on the same machine as the servers it connects.