Broadcast, UI and un-connected protocols - the future of Amateur Packet Radio?

Paul Evans W4/G4BKI, Tarpon Springs, FL.

Abstract

Anoverview of the user applications of packet radio is presented and the relevance of connected protocols in each scenario is examined. Suggested formats for unconnected systems are made and conclusions drawn.

Introduction

The increasing explosion of levels of data in the amateur network sometimes defies belief. Luckily, the amateur network (within obvious financial constraints) adapts at the local level to match each "crisis" in data throughput as it is reached. However, how long can this continue when spectrum is limited by bandplans and the necessity of keeping modes apart? Is the increasing use of ideologically "unsound" commercial wormholes for amateur use because of another reason? A deeper look at the efficiencies of spectrum use has to be taken and the inescapable fact is that protocols are the over-riding factor in channel efficiency, with modem design and TX/RX switching times probably being the next most important in turn.

Uses of packet radio

The vast majority of amateur packet usage can be sub-divided into the following applications:

1) Networking - end-to-end linking using ROSE, TheNet, TexNet, etc.

2) BBS - providing store and forward message service

3) PacketCluster - semi-real-time DX information and round-tabling

4) TCP/IP - robust connectless technology

5) APRS - real-time position and information system

6) End-user - keyboard to keyboard chatting

The order in which these applications are presented is a best estimate as to the amount of on-air time they occupy. Naturally, there is cross-fertilization between some of these areas - an end-user puts a message on a BBS and it forwards it to @WWW. The end-user did very little, the BBS "layer" did some more, but the network put it across the whole world - creating thousands or millions of packets in the process.

By way of further analysis, let's go through each one of these major applications to see where improvements could / should be made. The comments here are intended to be constructive and not cynical or demeaning of any of these applications which, after all, have appeared and adapted to suit each of their niches in turn. That is not to say that they have been implemented in the most efficient way, however.

1) Networking

The arena of networking is polarized into two schools of thought - ROSE and (the predominant) TheNet in it's various guises. As we know, the two work in distinctly different ways in routing data, with TheNet using the NET/ROM protocol of nodes broadcast in order to selectively adapt to changing RF routes. There is much to be done here in improving the network layer - such as using signal strength as part of the selection process in establishing path qualities. It is the author's opinion that ultimately, for better or worse, the nodes broadcast with embedded additional information in some form will win out in this area. Adaptive networks are the answer in RF data linking. ROSE does adaptation around "broken" paths, but not on the fly, it does it by selection of pre-programmed alternative routes.

Furthermore, it is increasingly common to see TCP/IP (sic. IP) data traveling on the ROSE and TheNet networks. Why? See TCP/IP below.

2) BBS

BBSes were the first real application of packet radio, after simple end-to-end keyboarding. Why did it start? You have to guess that it was because the typical amateur didn't want to burn telephone time talking to landline BBSes. However, much of the information by way of messages, files, etc. is repeatedly put over the air time after time in connected fashion to end-users. Monitor the traffic on the output of a BBS for long and you'll see this immensely inefficient use of spectrum. Similarly, the inter-BBS traffic has been pretty inefficiently handled. In the early days a more-or-less random forwarding system was used, at least nowadays hierarchical forwarding (in which the author was an instigator with G4MTP in the UK back in 1988) and data compression techniques are in place during forwarding.

3) PacketCluster

Several years after BBSes came PacketCluster, just when amateur BBS technology was maturing and obtaining a high degree of stability. Those of us who have a long term interest in HF saw this as the first "REAL" application of packet radio. It has to be said that the effects of PacketCluster have been widespread. Mainly because the data has to be timely, PacketCluster requires a fast network, with good reliable connectivity. Luckily, the owners of many of these systems have a lot of time and MONEY to invest with the final goal being only one thing - timely delivery of spots. Unfortunately, much of this investment is wasted because of the connected protocol used throughout the system. Just total up all the packets sent in the European or US-wide PacketCluster when one spot is entered. It is thousands. Multiply that by the number of spots entering the system during a contest weekend (probably one every 10 seconds) and it's no wonder that spots appear 10 minutes late - the whole network is creaking under the load.

4) TCP/IP

TCP/IP has enjoyed it's advocates (and critics alike) ever since it appeared on the amateur scene. The critics have mistakenly thought that it was a "channel hogger", stealing time from AX.25 This is not the case. It is simply a reflection upon the more efficient channel usage algorithm encoded in IP than in AX.25 Unfortunately, because TCP/IP is executed on an external PC and requires a higher degree of user input, there has been a mental picture that this must be for packet "gurus", with, let it be said, some resentment at times. When the protocol becomes closer to the average ham in the form of elements of the code in a TNC, then maybe this unfortunate distinction will no longer appear to exist.

The chief advantages of TCP/IP are:

a) it is commercially compatible. No other protocol in the amateur scene can so readily "hook" to DOS and UNIX applications

b) it uses connectless technology with (theoretically) nothing to time out and with on-the-fly re-routing. It IS efficient, because only frames get repeated over long distances if they are genuinely missing from a sequence

c) the protocol is very robust - it almost has to be by definition - it is not necessarily timely.

One of the difficulties of early usage of TCP/IP was building an entirely separate inter-connected TCP/IP network of stations. Now that ROSE and TheNet transport IP frames, the requirement for a different network has vanished. What is the most efficient long distance means of tranporting data now ? IP over network code.

5) APRS

Most amateurs in the USA will by now have heard of Automatic Packet Reporting System (APRS) software, written by Bob Bruninga, WA4APR. This consists of a full GPS position reporting network based entirely around UI broadcast frames. It's simplicity is the heart of it's functionality and possibly it's explosive growth in the past few months. Reporting stations give position reports or messages as UI frames (the messages have their own end-to-end acknowledgment protocol within the frame but are sent as UIs), while APRS stations periodically broadcast known targets in a similar way to TheNet. Provided there is a continuous end-to-end string of APRS stations, objects local to an APRS station in, say, Tampa can be seen hundreds of miles away in, say, Tallahassee. Every APRS station shares the data available.

The bottom line for APRS is that it is now a MASSIVE network nationally, and yet it still uses only one 2m frequency. Now that is spectrum efficiency!

6) End-user

Keyboard-to-keyboard chatting has virtually died it's death, maybe because hams are fascinated by talking to automatic gizmos. In fact, in some instances keyboarding which has been seen going on by a node Sysop has (mis-guidedly) resulted in those end-users being given limited access to the node! There are attempts in the South-East USA to produce an end-user network, but whether that will fully function with any great efficiency has yet to be seen. Keeping other traffic such as BBS forwarding and PacketCluster off clean, "pristine" bits o:f network is extremelv difficult to do (conversely, I know just how much money and effort went into keeping the UK PacketCluster backbone free of "unwanted" traffic). Unfortunately, with high loading connected technology, there will always be some element of antagonism between different user groups. At the end of the day, end-users are dependent upon some or all of the above systems to get their data through, unless of course it is simply one-on-one across town (in which case their packets could be lost on virtually any packet channel without distress or complaint by anyone).

Discussion

Possibly UI frames got a bad reputation because they were seen in the early days of packet as simply a means to provide Beacon frames. When users were desperate to see anything that was going on in t-heir area, beacons and use of the Mheard list were the only answer in AX.25. Very soon, however, the word was out that beacons were to be avoided at all costs other than at "key" network sites for identification. Their use fell dramatically at that point. Without question, the only use of unconnected streams from that point was TCP/IP.

The fly in the ointment that now raises great questions is, undoubtedly, the emergence of APRS. It has turned amateur connected technology thinking completely on it's head. Even with existing simplistic UI framing, APRS has proven that over a very doubtful HE path the positions of sailing yachts reported over an entire summer had a success rate of over 20% (and by extrapolation around 80% on VHF). The weakest area of APRS is having to work around existing non- optimized AX.25 UI protocol and that network code hasn't been able to transport end-user UI frames in connected format end-to-end. This is something that is sure to be resolved shortly.

It's clear that somewhere this channel efficiency was noticed by somebody. At the time APRS started it's growth spurt, around the InterNet went messages bemoaning the in-efficiency of PacketCluster. Why all those frames carrying the same message over and over to different addresses and often on the same frequency that the inter- node data came in on (horror of horrors, a flat data network!)?

Solutions

Clearly, we can see by now that there is a (very) common thread running through ALL of the above and ALL applications in the amateur packet environment. What's needed to fix the problems and how much will it help?

1) Use connect-less robust technology wherever possible.

2) At all possible network nodes, use the instantaneous RF route quality to reflect the true quality of the path(s). Evolve the most efficient means of dissemination of this routing information to neighboring nodes/switches.

3) Either change all inter-BBS traffic to broadcast addressed TCP/IP (255.255) or provide an alternative connectless piecing together of data frames on a shared basis (if I'm a BBS, I'll assemble as much of a message as it takes before I either accept or reject the message as something I will need to have or not).

4) All network code to have the ability to forward connect-less traffic transparently.

5) Modified UI frame to allow "ping" function. One frame is put out by a new (assumed to be highly mobile) user. Others answer in turn so that a complete local overview is built of channel activity.

6) Modified UI frame which allows information to either be broadcast in the "normal" fashion or addressed to a number of (listed) stations. Each station responds with an ACK. Retries are made by the "base" until the "delivery list" is clear or until time-out occurs.

7) Provide improved local network control using intelligent software with the ability to use either mobile GPS derived position information or previously broadcast fixed position information, in addition to RF routing quality information.

8) FEC type frames to provide more robust broadcast technology.

9) The ideal network is one which manages itself and adapts to suit the changing circumstances in which it operates. One reason why commercial and amateur networks fell apart after the 1994 Southern California earthquake was that either complete re-building of the network was required (skillful) or the latency of existing paths was too high. In times of system "panic" a heavy load "flood" algorithm should be available to re-establish routing as required.

Note: In future it looks like the position information of a station will become paramount in network control. There is an explosion in the use of Maidenhead locators being used by stations in UI frames (of the format [AAnnBB] with the square brackets). PLEASE, if you do send beacons for E's or tropo work, use this format.

These new "features" could have a very significant effect upon the requirement for data bandwidth. Let's look at a typical example:

As it is:

Take the situation where PacketCluster Node W6GO has 20 end-users and is linked on HF to VE3DXC with 15 end-users and the East Coast Cluster with, say, 20 cluster nodes and a total of 200 users (a not too uncommon situation).

User W6DU puts a single spot into W6GO over a two hop path. This gives 6 frames. W6GO then "creates" a minimum of 40 frames of activity with local users. Two frames (minimum) occur across the HF link. VE3DXC now creates another 30 frames locally and two more to the East Coast (in a DIFFERENT frame format too!). The East Coast cluster bursts into action and creates a total (minimum) of around 440 frames. In the end, 520 frames have been generated to support 235 users, NOT counting the numerous re-tries that there WILL be over all that distributed network OR taking into account any additional networking between nodes or from nodes to end-users.

As it "should" be:

User W6DU puts in the same spot. Users local to him use that information immediately. This one FEC frame (or 3 times repeated UI frame) tells W6GO there is a new spot. W6GO simultaneously sends one FEC frame which is heard by ALL users (either true end-users or the pseudo-user VE3DXC) IN THE SAME FORMAT. VE3DXC immediately does the same single frame FEC to his end users, etc. The number of frames is therefore reduced to a minimum of 25.

The result: a 20 fold decrease in spectrum occupancy of all types. Just stop and think. Why did you just spend all that money going to 9600 Baud dedicated PacketCluster links when 1200 Baud will do it easily? Because the protocol used forced you to. Just think what 9600 Baud will REALLY do for you.

Adding this new data efficiency in the real world.

This could be done two ways. Firstly, build a new special version of code to implement just the new protocol (potentially useful for dedicated, high speed sites). Secondly, build expanded code which would fit in with the existing network, providing a new "super-set" of functions. To do that it must first be built around what's out there in use - the TNC-2 clone. One problem which has dogged the code writers at all the TNC-2 manufacturers has been one of CODE space. For some time now, modem TNC-2 code really has been a quart being squeezed into a pint pot. This is an area where no single TNC manufacturer has been prepared to boldly break the mold of TNC-2 compatibility on the Z-80 platform, and for some very good reasons. Fortunately, the network code authors (notably Dave Roberts, G8KBB, author of TheNet X1.52) have provided a standardization route here. The most frequently used alternative code in a TNC-2 is network code and so it is logical that anything that works there is workable for all. The development of the bankswitch (using a firmware driven SIO line to switch between the lower and upper page of a double size EPROM) has suddenly extended the life of the Z80 TNC-2 and all of it's compatibility way into the future. This now makes available all that code space which can be used to provide new code power as well as backwards compatibility with "pure" AX.25. The way forward appears to an ad-mix of some elements of network code, IP and end-user code residing within the end-user TNC. Digipeating was always in the end-user code, why not move more level 3/4 "smarts" into the end-user TNC and build routing tables and connectless protocols on the expanded platform?

Faster processors in the TNC-2 platform (such as the new PacComm 20MHz processor Backbone version of the SPRINT-2 TNC) will further allow more exotic code to be run before the TNC-2 design runs out of life.

Future platforms will certainly have to be built around next generation micro-processors which are ideally suited to real-time data I/O tasks, either the 28018 1/2 or 68000 series. Much of the existing Z80 code could be used with the 280182, whereas 68000 code would need to be built from scratch (although the Gracilis route from TCP/IP and AX.25 compatibility in NOS is certainly attractive here).

Despite these dreams, the TNC-2 compatible, it can be assumed, will be with us for a long while yet and that it where the code will probably be targeted in the first instance.

Has this realization process been started?

The answer, to limited extent, is YES! The process was started in a discussion between the author and WA4APR in which the amount of data emanating from PacketCluster was mentioned and so were some ways in which more generalized improvements in data transport might be realized. It has to admitted too that many commercial applications would benefit from such improvements. Very soon Bob had incorporated code into APRS which would monitor PacketCluster traffic, throw away unwanted information, monitor inter-node traffic, etc. Basically give you information the moment it becomes available (which, you'll recall, is what it's all about,). Expanding upon this, stations including Maidenhead locators where instantly placed on a correctly scaled map. Finally, the DX spots themselves are placed on the map as they come in.

Future improvements may be to create multi-user "switches" using connectless technology and to use position information for network building and control . The applications are general, but PacketCluster is certainly going to be one of the chief beneficiaries of this approach.

Furthermore, the TNC code side is currently under development to provide enhanced end-user connectless technology. Once these areas of supervisory (IBM-PC) code and TNC code are more fully developed, one can expect it to start displacing or molding the design of existing systems around it.

The bottom line

Existing data dissemination systems deliberately steer away from shared data. Why is APRS so efficient? It works around a COMPLETELY flat, data SHARING system using ONE channel. It really does make you think "where did we go wrong?".

Conclusion

Without some major re-thinking of protocols, the amateur network will forever fight un-necessary data bottlenecks. For the foreseeable future, data rates will be limited by the availability of suitable radios. Until the day that high speed radios are readily available, the main improvement to the network will be made in the area of protocol. More specifically, protocol which has been adapted to the needs of the applications most favored by the end-users. As practically all the systems provide one-to-many services in one form or another, this is the area where work would be best focused. There is still a place for connected technology, but more and more it looks like unconnected technology is the future of amateur packet.