Packet Radio at 19.2kB -- A Progress Report

by John Ackermann AG9V
john.ackermann@daytonOH.ncr.com
Miami Valley FM Association, Dayton, Ohio

Overview

This article briefly describes the technical aspects of the 19.2kB backbone and metropolitan area network (MAN) now operating in Southwestern Ohio.

Although there has been a lot of discussion of fast packet radio systems -- from 56kB on up to 10MB -- in the packet world, the fact remains that ten years ago we knew that 1200 baud systems were only a stopgap and that we needed to move to higher speeds before packet would come into its own. Today, there is still a dearth of equipment commercially available to operate at speeds greater than 1200 baud.

When Phil Anderson of Kantronics first started talking about his plan for a transceiver that would do 19.2kB out of the box, a group of hams here began to consider the possibility of building a high speed network to link the area's population centers, and to provide high speed user and server access as well.

The Kantronics equipment doesn't represent the state of the art in fast bits. But, it *is* the first system that allows a major improvement in packet performance using off-the-shelf components designed for the task. To put it simply, the people funding the Ohio network wouldn't have put up their money for the sort of do-it-yourself approach that other high speed packet systems have required.

Their goal was to build an infrastructure to support the general amateur community, not a plaything for tinkerers, and off-the-shelf equipment is what finally convinced the ham community that this system might actually work.

The clubs and individuals involved, the Miami Valley FM Association, the Central Ohio Packet Association, the Ohio Packet Council, K1LT, N8XX, and AG9V, ended up building a backbone linking Dayton, Cincinnati, and Columbus, Ohio through a total of five network switches. The backbone currently has about 150 miles of 19.2kB radio links and links to the existing Ohio 4.8kB network at Columbus.

Additionally, in Dayton the MVFMA and AG9V have built a MAN around a full-duplex repeater to provide a hidden-transmitter free network at 19.2kB for end users and servers. The MAN is tied to the backbone through a switch?

RF Hardware

The Kantronics D4-10[2] is a UHF radio designed expressly for data communication sew--ices. It is rated at 10 watts output, though our tests show that most radios put out about 12 watts. It has two crystal controlled channels and by default comes from Kantronics with crystals for 430.55MHz installed. The receiver is a completely different design than the earlier 2 meter DVR2-2 radio, and has much better spurious signal rejection

The radio has selectable receiver bandwidths with a nominal 15kHz IF for voice and low speed use, and a 60kHz bandwidth for 19.2kB.

---

[1] I'll use the term "network" to refer generically to the backbone and MAN unless it is important to distinguish between them.

[2] Since this article deals almost exclusively with Kantronics equipment, I should note that none of the folks involved in this project are affiliated with that company.

The D4 supports normal analog signal inputs and outputs, but also has a TTL level I/O port which is intended for 19.2kB use. The TXD line on this port drives an FSK circuit that shifts the transmitted frequency +/- 10kHz from center for mark and space. The radio provides pulse shaping to meet FCC bandwidth requirements at 19.2kB, so the TXD signal need not be processed before feeding it to the D4.

The RXD line comes from the output of a comparator that acts as a data slicer driven by the discriminator. DCD is driven by the D4 squelch, and PTT is a normal ground-to-key signal.

Turnaround time is very fast. Using a squelch-derived DCD signal, two of the radios will talk to each other with a TXDelay of 5 milliseconds, though using 10ms or so probably is safer (these tests were done using the Ottawa PI card, which offers 1ms timer resolution).

We've had several of the radios in service for five or six months and have had no failures. We have seen some long-term frequency drift but nothing beyond what would be expected with 6 MHz crystals multiplied to 430MHz. Apart from one radio that has a serious stability problem over very small temperature excursions (we suspect either a bad component or a bad crystal), so far temperature stability has not been a problem.

A pair of D4s talking to each other over moderate paths will tolerate a bit more than 4kHz of frequency error. We recently learned that Kantronics has come up with a couple of modifications to both improve frequency stability and increase the amount of tolerable frequency error; with these mods the radios are supposed to tolerate about a 6kHz frequency error. We expect to have more information on these modifications soon and will be installing them in our radios.

The only adjustment we've found to be somewhat critical is the threshold setting for the RX data slicer. This adjustment sets the point at which the slicer shifts from outputting a 0 to a 1 and ideally should be set to cause that transition at the center of the channel.

We've seen a couple of radios where this adjustment (R17 -- a pot that sets the DC reference signal to the comparator) was out of adjustment. As a result, the radios wouldn't decode signals from an on-frequency transmitter, but would do just fine with a signal that was off-frequency in the direction of the threshold error. One of the mods referred to above supposedly reduces the touchiness of this setting.

Adjusting the threshold requires an on-frequency signal with tone modulation and an oscilloscope. Once the threshold is set, it seems to stay put. We think the problems we saw were due to initial misadjustment or to the banging around some of the radios have seen; none of the radios have required a second tweak.

The D4 works well in high RF environments. The MAN repeater antenna has several other commercial and amateur VHF/UHF systems in close proximity, and at his house Lew, K0RR, runs 100W on 435MHz (OSCAR uplink) with only four feet of separation from the Isopole that's hooked up to the D4 on 430.95. He receives packets without error during uplink activity.

RF Paths

We were a bit concerned about whether 10 watt radios operating in a 60kHz bandwidth would have enough power for the longish paths we needed to span. We have found that as long as the RF path is line-of-sight (or close to it), the D4's power is sufficient for paths of up to at least 50 miles.

But that "LOS" qualifier is critical. Even on relatively short (10 mile) paths, if the antennas aren't in the clear and above the foliage at both ends, the path won't work. This was driven home during our testing this spring, when AG9V was trying to access the MAN repeater, which was at an excellent location, but with the antenna only about 20 feet above ground.

AG9V is about 10 miles away and uses a 22 element K1FO beam up about 45 feet, but in a rather low and heavily forested area. Before the foliage came out, the path worked moderately well. Once the leaves were on trees, however, signals *totally* disappeared. Raising the repeater antenna to its final height of 120 feet brought back rock-solid signals.

Most of our point-to-point links use Cushcraft 11 element 435MHz yagis on 5 foot booms. We've found that surplus 3/4 inch 75 ohm CATV hardline works well. The slight mismatch doesn't seem to cause a problem on either our simplex links or on the repeater antenna.

Modems

We are using two different (very different) modem schemes. The backbone, which presently uses Kantronics DataEngines running G8BPQ node software (see below) uses the Kantronics 19k2/9k6 modem. This is essentially a G3RUH design with doubled clock rate and all the analog parts bypassed; the output of the scrambler is fed into the D4 TTL port.

On the MAN, we're using a different approach. We are feeding HDLC frames directly into the D4's TTL port without using a scrambler or any other modem components. Essentially, we are treating the D4 as an RF modem. There have been many theoretical discussions about whether the scrambler is necessary to provide reliable clock recovery and demodulation of NRZI FSK signals, but our real-world experience indicates that using the K9NG/G3RUH scrambler as implemented in the Kantronics modem adds no significant benefit.

We have a 35 mile path operational without a modem, and it seems to be just as reliable as similar paths using the Kantronics modem. (It's been pointed out that the scrambler also allows use of a tracking data slicer, but since the Kantronics 19k2 modem uses the D4's internal slicer, that's not an issue here.)

Using the modemless[3] approach requires that DCD be derived from the D4's squelch. This isn't a performance problem, as the D4 squelch is very fast and solid, The 5ms turnaround time noted above includes squelch response time so it's apparent that there's no speed penalty in this approach. I'll leave for another time the argument about whether a system should recognize only data, or any signaf, on channel as DCD. Note, too, that because this configuration doesn't scramble the data: it will not talk to 'RUH modem-based systems.

Bit Bangers

Several different devices have been tested as packet assemblers and HDLC generators.

We have used the Kantronics DataEngine both with the 19k2 modem and our "modemless" configuration. We have several DEs in place running the G8BPQ network code and they do just fine supporting two 19.2kB links on the radio ports and a couple of slower links on the RS-232 port.

The DataEngine has one problem -- if it is connected to high-speed modems that allow the RXD line to chatter when there's no signal present, the interrupts generated will saturate the V40 processor and prevent it from properly servicing the serial port. This isn't very noticeable when running the standard TNC2 command set on the DE, but it is very apparent when running KISS mode, or using the G8BPQ node software.

Kantronics says they're working on a software fix for this problem. A relatively simple hardware mod can tremendously improve things -- simply gate the modem's RXD output with DCD so the chatter doesn't make it through to the DataEngine. We've made this change at our G8BPQ switch sites, and with it the DataEngine works well.

To use the DataEngine to feed HDLC directly to the D4, we use an internal jumper board (available from Kantronics for about $20) to provide the proper signals to the DE's radio

[3]Technically, the D4 is acting as an RF modem when it's being driven with digital signal levels. I use the term "modemless" to contrast with systems using a separate device between the output of the PAD and the input to the radio.

port. One bit of hacking is required: you need to add a 4020 or similar chip to divide the 16x clock the DE provides down to 1x clock, and feed that back to the TNC. The mod is no big deal; the chip fits neatly on the jumper board in "dead bug" style. Again, gating the RXD line will probably make the DataEngine happier.

A TNC with speeded up clock and components will also work well at 19.2kB. We've put three of PacComm's Tiny-2 TNC2 clones with 10MHz crystal and CPU[4] on line with no problems and full interoperability with our other HDLC generators.

By doubling the crystal speed, the baud rates are also doubled, so the 10MHz Tiny-2 will support 38.4kB on the serial port and 19.2 on the radio port without any further modifications. The interface to the D4's TTL port is nothing more than a cable from the appropriate pins on the modem disconnect header.

One minor annoyance is that the PTT line is not included on the Tiny-2's header. You can either hope that the SIO's RTS pin will sink the D4's PTT current, or pick up PTT from the Tiny-2 rear panel radio connector.

The Ottawa PI interface card is theoretically the best PAD to use at these speeds because its DMA data transfer can handle higher speeds with less CPU load than a PC serial port. K0RR and AG9V used PI cards for their first experiments with the D4 over short-haul links.

In the real world, the PI card works well for us, but in ping testing we see a troublesome loss of 5 to 10 percent of the packets sent even over near-perfect paths. We don't see this loss when using Tiny-2s. Dave Perry, VE3IFB, has revised the PI driver software for NOS, and that provided some improvement, but the problem still persists. This packet loss is almost certainly the result of some sort of timing glitch, but as yet we haven't tracked it down.

[4]The 10MHz "node version" of the Tiny-2 is available from PacComm for about $25 more than the standard model.

In short, we'd like to use PI cards in our NOS-based systems and switches, but we are being cautious until we fix the dropping pings. The Ottawa folks have been very helpful, and I'm sure we'll get this licked soon. (This problem is unique to our setup with the D4 radios; others are using the cards with the 56kB GRAPES modems and reporting excellent results.)

The other problem with the PI card is that it doesn't work with computers other than PCs, or (at least for now) software other than NOS.

In summary, at the moment the best price/ performance ratio in bit generating hardware is the 10MHz Tiny-2 ($150) coupled with a 16550AFN UART in your serial card. The PI card ($120US) will be the best answer for PC NOS users once we get the timing bug worked out.

Repeater

The Dayton MAN is built around a full-duplex repeater. The repeater ensures that every station can hear every other station, and helps reliability by allowing use of high-gain yagis pointed at the repeater site.

The repeater hardware is nothing special; it's simply two D4 radios connected back to back through their TTL ports. A 4049 CMOS inverter provides buffering. The only trick we learned is that since the D4 uses op amps rather than true TTL devices to drive the TTL port, hooking a pair of radios directly back-to-back won't work. You need to use some judicious pull-ups and pull-downs to get proper signal levels.

The current controller has nothing more than the interface circuitry, a 15 second time-out timer, and a control function to drop the transmitter. The next generation will add appropriate ORing circuits so that a switch can use the repeater as a radio port. That will tie the repeater to the high-speed backbone.

We also want to add a bit regenerator to help the quality of the transmitted signal. We'll do this because it's the right thing to do, but at the moment it doesn't appear that we are losing many frames due to distortion in the repeater.

The repeater uses no squelch tail; PTT is driven directly by DCD. The D4 squelch and keyup time are fast enough to permit a TXDelay of 15 to 20 milliseconds through the repeater.

The two D4 radios are connected to a small TxRx duplexer (four cavities with a little band-pass and a lot of notch filtering) and the duplexer feeds an 11.5dB gain Diamond fiberglass antenna through about 160 feet of 3/4 inch CATV hardline. The antenna is about 120 feet up at a site that towers (by as much as 100 feet) over the rugged Ohio skyline. Frequency separation between receive and transmit is 10MHz (420.95 in, 430.95 out) and we haven't noticed any desensitization.

There are currently six stations using the repeater, with paths ranging from about 1 mile up to 35 miles. The DataEngine, Tiny-2, and PI card are all being used to generate packets, and all three interoperate with no trouble. The repeater carries BBS, TCP/IP, NetRom, and PacketCluster traffic.

Switch Hardware and Software

At the moment, our nodes primarily use G8BPQ code running in DataEngines. We hope to change over to NOS-based switches that can handle both IP and NetRom switching. Johan Reinalda, WG7J, has NOS running on the DataEngine, and we plan to start experimenting with that code soon.

Apart from the inherent limitations of the NetRom protocol, the G8BQP/DataEngine combination works very well, and we've seen switches run for months without a reset.

Frequency Allocation

Finding space for a bunch of 100kHz wide channels on the 70cm band isn't easy. The ARRL-recommended segment of 430-431 MHz happens to fall in the top of the 426.75MHz ATV channel. Among other things, this means that an ATV transmitter puts its audio subcarrier at 430.75; a local ATV repeater uses that channel for its output and that rude surprise required us to change the repeater frequency shortly after we started testing

Apart from ATV, the 420-430 segment of the band is used in Ohio for voice links, many of which are uncoordinated.

After a lot of experimentation and negotiation (and with tremendous cooperation from DARA -- the Hamvention folks -- who agreed to move a bunch of their planned voice links for us) we wound up with the 420.5 to 421.0MHz range free for our repeater input and point-to-point paths. It appears that we can coexist with the ATV signal in the 430.5 to 431.0 segment if we stay clear of 430.75. We haven't tested yet, but we suspect that operation below 430.5MHz may result in interference with/to the video signal.

Kantronics ships the D4 by default with 430.55MHz crystals installed in channel 1; we got all our radios with that channel and use the second channel position for our operating channels. We don't have any links assigned to 430.55, so it is available for temporary and testing use and we can grab any two radios and make them talk to each other. We recommend that others follow this practice and reserve 430.55 as a testing and experimental channel..

The long and short of it is that frequency coordination will be a prime concern for users of D4 radios in populous areas. Living with ATV is likely to be the biggest challenge.

Acknowledgments

Primary funding for the backbone was provided by the Miami Valley FM Association, the Central Ohio Packet Association, the Ohio Packet Council, and Hank Greeb, N8XX. The Dayton MAN was funded by the MVFMA with minor contributions by AG9V. NCR Corporation kindly donated computer equipment that was surplus to them but critical to us; that equipment is be-

ing used to provide applications across the network.

Vic Kean, K1LT, is the technical guru for the project. He burned the ROMs, tuned the radios, arranged for the sites, and climbed the towers. Without Vic, this network would not exist.

Kantronics provided us with the pair of beta-version radios that got the whole project started, and with technical support along the way.

Finally, the support of KORR, who bought the second PI card in town, and N8KZA, N8ACV, WB8GXB, and K8GKH, stalwarts of the MVFMA, was invaluable in getting the Dayton end of things running.