PRIORITIZED ACKNOWLEDGEMENT (PRIACK) PROTOCOL

Eric Gustafson, N7CL

For the last several years I have been trying to advocate a very minor change to the AX.25 protocol. This change will allow the protocol to be much more compatible with our multiple access half duplex radio channels. I requested that the digital committee consider including this change in the protocol specification. The committee indicated that the idea might have some merit. They suggested that I get some test code written to try it out on the air. Since that time, I have worked with Howard Goldstein, N2WX to get some beta test code written so that this prioritized acknowledgement (PRIACK) channel access system could be evaluated on the air. What follows is a very non-rigorous description of the modified protocol.

While primarily intended to improve channel performance on HF, this system will also be useful on any VHF simplex channels where there are hidden terminals.

BETA test code is now available from me for anyone who is genuinely interested in participating in the test AND REPORTING THEIR FINDINGS. I have no interest in simply distributing ROMs and then never receiving any feedback from the user. Two things are needed at this time. First, reports of bugs or protocol violations would be most useful for tuning the firmware so that the protocol modification itself can be meaningfully evaluated. Second, a valid quantitative test of the effect of the protocol modification on a multiple access HF channel needs to be done. The code I have is for TNC-2 clones and the MFJ-1278. I can be reached at:

Eric Gustafson, N7CL
2018 S. Avenida Planeta
Tucson, AZ 85710
(602)-747-1410

I would prefer to deal directly with area test coordinators on Compuserve. Those interested in becoming an "area test coordinator" should contact me in section 9 of the HAMNET forum on CompuServe. Use a subject keyword of "PRIACK". My CompuServe number is 71750,2133.

I had originally intended to report the results of testing done at HF and VHF in this paper. The initial results of VHF testing done here locally on two simplex LANs is very encouraging. Even with small numbers of users on the channel the improvement was quite dramatic. Putting the code into the TNC of a BBS which was on 145.01 (along with users and node inputs) had the effect of making the channel functional again for the users. Previously, whenever someone accessed the BBS, all other activity on the channel effectively came to a halt. So far we have noted no instances of incompatibility with the old protocol.

Unfortunately, HF testing has been slow to get off the ground since both of the initial beta test groups have a large contingent of members who are using PK-232 TNCs. Until recently this meant that these members couldn't participate in the test due to not having PRIACK code available for the PK-232 This has now been corrected. AEA has recently made some beta test code available for the PK-232. I understand that AEA is planning on including PRIACK in an upcoming release of generally available code. Hopefully, we can now do a meaningful test of the PRIACK system on HF.

DESCRIPTION

The idea behind the prioritized acknowledgement (ACK) protocol is quite simple. The idea is to give ACKs priority access to the channel so that time is not wasted retrying packets that have already been correctly copied but for whatever reason, the ACK is not received within the time limit defined by the FRACK (T1) timer.

The present protocol does not handle a simplex LAN with hidden terminals as well as it possibly could. This is primarily because, the present protocol is more likely to synchronize collisions with acknowledgement packets than with any other type of packet.

To this collision synchronization mechanism the current version of AX.25 adds a propensity to cause even ACKs which are not from hidden terminals (and therefore less susceptible to collision) to be delayed beyond even generous FRACK timer settings when the channel gets busy.

Once the FRACK timer times out, even if the ACK finally makes it through before the retry is sent, the original packet is retried anyway. This obviously wastes a lot of time which could be better used clearing the channel of some of the legitimate offered load.

It is this feature of the current AX.25 protocol that accounts for most of the abysmally poor performance of the currently popular NETROM and THENET nodes when they are used as omnidirectional single channel (or even multichannel if there is more than a single other node on each channel) systems. Poor receiver not ready (RNR) condition handling by the current protocol is also a contributor here. It should be noted that these node chips CAN handle point to point links to a single other node perfectly adequately.

The prioritized ACK protocol avoids the above problems by giving ACKs priority access to the channel. It does this in such a way that even ACKs coming from hidden terminals are protected from collision. The current protocol gives a limited version of this priority access only to digipeated frames.

ACK prioritization works with slotted channel access in the following way:

1. Response frames (ACKs) are always sent immediately with no time delays unrelated to hardware limitations applied. Ultimately, not even data carrier detect (DCD) should be checked for sending an ACK. However, in the current beta test code, DCD will still hold an ACK off the channel. This was necessary to allow compatible operation with non PRIACK stations.

2. Stations queued up to access the channel but waiting for a channel busy condition (DCD true) to clear, will start a slotted access procedure only AFTER enough time for a response frame to clear the channel has transpired (whether or not the response frame is detectable by the queued up station).

3. Slot time windows are selected to be large enough that the local TNC will be able to unambiguously determine whether any other detectable station has selected any slot preceding the slot selected by the local TNC. This is to prevent two TNCs which have selected adjacent slots from colliding anyhow.

As you can see, under this protocol there will never be a condition where an ACK is delayed from being sent beyond the FRACK timer limitation. In fact, the FRACK timer becomes relatively meaningless in this context. However, the FRACK timer is still required to maintain compatibility with stations who are not running PRIACK code. When all station are running the PRIACK system, the TNCs know that if they don't see the ACK immediately when expected, they are never going to see it.

FRACK can be used to force delayed reaccess to the channel when an ACK hasn't been received when expected. However, when all stations on the channel are running PRIACK, both FRACK and calculated backoff types of access control are completely unnecessary. Here I am assuming that end to end AC digipeating will be phased out of the protocol specification now that networking development has begun. If layer 2 digipeating must continue to be supported (yech!), we should at least convert to point to point ACKs for this purpose.

On a marginal channel which is lightly loaded, calculated backoffs will do exactly the wrong thing by backing off reaccess times on the assumption that nonreceipt of ACKs is due to collisions. Under the same circumstances, PRIACK will try to get through whenever it isn't deferred by DCD being true. If the channel truly IS busy, PRIACK equitably distributes the channel bandwidth across all the users. This fair distribution is primarily a result of using p-persistence slotted channel access control ala Karn and Lloyd [1]. It is the combination of prioritized acknowledgements and carefully structured slotted access control that makes PRIACK so effective.

Enforcing a channel access delay for all stations on the channel for whom the packet that caused the queue was not intended (and who therefore aren't going to ACK it) allows even ACKs from hidden terminals to get back to the expecting station. This clears that traffic from the offered load list. If the packet was indeed copied and ACKed, further retries of the same information will not be necessary.

The beta test code for this modification to the protocol is compatible with stations using the current protocol. A station using the new protocol will not degrade the channel for users of the current protocol. So there is nothing wrong with firing up the new stuff on a channel where the majority of the users aren't yet using it. Several of us here in the Tucson LAN have been running PRIACK exclusively for over nine months now. Other stations on the LAN are (so far) completely unaware of this fact.

HARDWARE LIMITATIONS

This protocol will not work well if the TNC's modem DCD characteristics are not very good, but neither will any other CSMA based protocol.

If the TNC uses the 2211 demodulator, the DCD characteristics can be readily optimized. A relatively simple modification to the DCD circuit of these demodulators is all that is required to make the DCD reliable enough to base a CSMA system on.

If the TNC uses one of the single chip modems intended for land line use (like the AMD7910 or TMS3105) or a filter based modem like the PK-232, all is not lost but a much more extensive modification is required to rehabilitate the DCD circuit.

The information needed for both of these modifications is in my paper in the proceedings of the Seventh ARRL Computer Networking Conference [2]. In addition, TAPR has made circuit boards available which eliminate hacking the TNC circuit board in the case of the 2211 and eliminate the tedious state machine wiring task in the case of the other modems.

In particular, if your DCD circuit has a high false DCD duty cycle when listening to noise on an empty channel, the protocol's ACKWAIT characteristic will virtually prevent you from being able to transmit at all. The DCD circuit MUST be able to reliably detect the presence of a data carrier but it MUST NOT be constantly chattering away on noise.

TERRESTRIAL APPLICATION ONLY

The PRIACK system is applicable only to terrestrial communications and only in the half duplex radio environment. Long path delays, ALOHA channels, and full duplex point to point links all have different requirements and should have protocol specifications tuned to the specific application. Most of our layer2 problems have come from one of two sources. One is trying to incorporate higher level functions in the layer 2 specification (CHECK etc.). The other is from trying to make one layer 2 protocol fit many differing layer 1 topologies. Unfortunately, what is going on at layer 1 DOES affect the way the layer 2 protocol should be structured.

CONCLUSION

If we are to optimize our amateur radio protocols, we will have to tune the protocol specifications to optimize efficiency based on the layer 1 limitations specific to the application. I believe that the PRIACK system is a necessary protocol tweak for use on terrestrial multiple access half duplex radio channels.

Special thanks to:

Howard Goldstein, N2WX, who coded these protocol modifications into the TNC-2 AX.25 code so that we can begin testing with enough stations to make a useful statement about performance.

Martin F. Jue, and Steve Pan of MFJ Inc. for allowing me to release some 1278 V2.3 EPROMs with this modified protocol installed. This will be a very big help in testing on HF where the need is greatest.

AEA Inc. for devoting the resources to providing PRIACK code for their PK-232. This will be a big help in getting a meaningful HF test accomplished.

Lyle Johnson, WA7GXD, who has contributed many hours kibitzing about the various aspects of the protocol and considerable time participating in the preliminary on air testing.

Dan Morrison, KV7B, whose contribution has been similar to Lyle's.

Gail Peterson, N7BXX, who has been helping out with the preliminary on air testing.

Mykle Raymond, N7JZT, who has been helping out with the preliminary on air testing.

References

[1] Karn, Phil, KA9Q and Lloyd, Brian, WB6RQN, "Link Level Protocols Revisited," Proceedings of the 5th ARRL Computer Networking Conference, pp 5.25-5.37.

[2] Gustafson, Eric, N7CL, "Can We Continue To Ignore Level One?," Proceedings of the 7th ARRL Computer Networking Conference, pp 74-82.