"HOW TO STAY CONNECTED TO YOUR PACKETCLUSTER NODE"

PacketCluster Parameters from a User's perspective

by Jay O'Brien, W6GO

Preface:

The following paper is prepared for the WIMU (Wyoming, Idaho, Montana, Utah) convention, August 5, 1995, in Jackson, Wyoming.

I was asked by Wayne Mills, N7NG, to talk about what a PacketCluster user could do to better utilize a PacketCluster node (Wayne is a node SYSOP in addition to being a world-class DXpedition operator).

What are my qualifications for presenting this paper? My PacketCluster node went on the air on April 20, 1988. I was a principal beta tester for several versions of PacketCluster and many of the PacketCluster features are there at my suggestion. My node now has five other nodes connected directly via backbone connects and as many as 4 nodes connected in addition on 10MHz. My node has direct user connects and user connects via two "X1J" network nodes. My node has has as many as 40 user connects at one time during a contest. Assisted by others who are technically aware of AX.25 packet protocol, I have worked with many users to identify and correct user connect problems, and in all cases
except one we have been successful in resolving the problems.

SUBJECTS TO BE COVERED:

- A node sends a DX spot to a user. What really happens?

- A node sends a DX spot to a LOT of users. What really happens?

- A TNC has a lot of PARMS. What are they, how do they keep you connected?

Note: RF problems are not covered here.

1. ANATOMY OF A DX SPOT:

When a PacketCluster node sends the following spot, what happens?

DX de W6GO: 14020.3 5A1A QSX 026 wants W6/7 0550Z

If there is only one user on a clear RF channel, here's the sequence:

Timing of a DX spot sent to a user (times shown are in milliseconds):

NODE TRANSMISSION (957.5 ms):

0.25	Transmit process time
250	Transmit turn on time (TXDELAY)
6.67	Flag byte (always 01111110 binary)
137	Callsigns and SSIDs, average 20 chars (range is 14-70 characters)
6.67	Frame type byte (signifies this is an information frame)
6.67	Protocol Identifier byte (changes if between NetRoms)
515	DX spot. 72 characters plus two bell characters, one CR character
13.33	Frame check sequence (used for error checking)
6.67	Flag byte
15	Data receipt allowance (carrier hang on)
.03	Propagation delay
.25	Receiver process time

USER ACKNOWLEDGEMENT TO NODE (435.9ms):

.25	Transmit process time
250	Transmit turn on time
6.67	Flag byte (always 01111110 binary)
137	Callsigns and SSIDs, average 20 chars (range is 14-70 characters)
6.67	Frame type byte (signifies this is a supervisory frame)
13.33	Frame check sequence (used for error checking)
6.67	Flag byte
15	Data receipt allowance (carrier hang on)
.03	Propagation delay
.25	Receiver process time

1393.41 ms total, or 1.4 seconds.

Conclusions:

1. Information transmitted was 75 bytes, and it took 1.4 seconds.
2. The data transmission rate is 53.6 bytes per second, or 428.6 baud, not 1200 baud.

Assumptions:

1. No delay imposed in sending acknowledgement to node.
2. No retries necessary due to QRM or signal fade.
3. 3% bit stuffing by AX.25 algorithms which prevent 6 consecutive one bits anywhere other than in the flags.

2. WHAT'S WRONG WITH THIS EXAMPLE?

If a DX spot takes 1.4 seconds, then a user should be able to get 42 DX spots a minute, right?

Right, if there is ONLY ONE USER connected to the node, the user has a perfect connect, the user frequency has no other traffic, and the node and user are adjusted to capitalize on this perfect situation.

The real world has LOTS of users on the frequency, perhaps some Network nodes relaying users to the node, and maybe other traffic like Cluster users from another node on the same frequency or non-cluster traffic. What happens then?

To reduce the average time for each DX spot, the node will key up and send the spot to 5 or more users, thus using the TXDELAY time only once for all of the spots. Then, the node will switch to receive and collect acknowledgements from users. When the frequency is clear the node will send the spot to another group of stations. Some of these stations will transmit on top of each other and will then resend their acks. It is a mess!

Think about it. If there are 20 stations connected and there are no collisions, when a DX spot goes out the node transmits about 20 seconds to send to everyone, and if everyone responds politely, it takes ten seconds to get all of the acknowledgements from the users.
That implies it is possible to send two spots a minute. Actually, this IS possible! But only for short bursts.

How do we get around this problem? We tell our TNCs to be polite and passive, minimizing arguments between users of the frequency. We tell the TNC how to act using directives called "PARAMETERS", or PARMS. More on PARMS later. Right now, a bit more description of the problem.

Tom Wood, N6IXX, wrote the following discussion some years ago. I have edited it a bit to bring it up to date. It still applies and will help to explain.

"This is a discussion about TNC parameters and what they mean to the typical PacketCluster user. This was written with the knowledge that most PacketCluster users are new to packet, and of the belief that this "parameter stuff" is too complicated for them. Trust me, it's not.....

"If you will change a few of the parameters in your TNC, you will, without question see an improvement in your PacketCluster operation.

"The PacketCluster node is an anomaly in packet radio. PacketCluster operates as a real time spotting network. What this amounts to is an RF Riot when a DX announcement goes out. The parameters in use in the "regular" packet world cause problems here.

"Why ??

"Picture the following: 20 people are asking non-stop questions, and they are all asking them at the same time. Unfortunately, there is only 1 person to answer them. As a result, the 20 people asking the questions have to scream, yell, and repeat their questions again and again. In a situation like this there is not much information being exchanged ...

"Now picture this: All of these people are in a very large room. Some of the 20 question asker's are so far away from the answer person that they cannot be heard. So they enlist the help of a "relay" person....they pass the question through the "relay" person who sends it on to the answer person. Now you have 20 people AND a few "relay" people asking questions ----> of one person (!!).

"If everyone would simply slow down and wait their turn, things would go much more smoothly. Ask a question, get an answer. Ask a question, get an answer.

"Where am I going with this ??

"The scenario I have outlined above is almost exactly what happens on PacketCluster. Almost all of the TNC's manufactured today are programmed with the EXACT same parameters. Most people buy the TNC and put it on the air with no regard to parameter settings. The result is that these people are pounding each other in an attempt to communicate with a PacketCluster node.

"Add an X1J or NET/ROM Node and you now have the "relays" mentioned above. All of these devices are competing for the "ear" of the PacketCluster Node.

"With everyone's TNC set to behave in the same manner, there are collisions all over the place, throughput is reduced and everyone pays the price in "sluggish" operation.

"By adjusting some key parameters in your TNC, you set yourself apart from the crowd. You transmit when they don't. The Node hears you after only 1 or 2 tries. When everyone adjusts their parameters, they are in effect taking turns at communicating with the Node. Information flows more smoothly. And things actually run at a faster clip.

"It is particularly important to adjust your TNC parameters when a X1J or Net/Rom node is being used. Under these conditions, someone is transmitting almost all of the time. And, you don't hear many of the other users, compounding the problem.

"If things don't work perfectly, try minor PARM changes. You may be conflicting with another user. If everyone uses the same PARMS, that's a problem in itself! Some "fiddling" with TNC settings may be in order. Change a little here, change a little there. You will notice a difference with each change. Yes, you might even have to dig out the 'book'."

N6IXX spent untold hours watching his monitor screen before he wrote this paper. He was right then, and he is right now. His work, along with input Tom received from WA8DED, is "the book" on sharing a PacketCluster frequency.

I would like to add these thoughts to to Tom's work:

- All the PARM settings in the world won't fix a bad RF path.

- User PARMS have NO control over how fast a node sends spots. Only the node PARMS have any effect on the distribution of data from the node.

- Agressive user PARMS will make the node appear "snappier" when the user is sending something to the node, but those same agressive user PARMS will cause both the agressive user AND OTHER users to be disconnected when there is no reason for the disconnect.

What to do? Tell your TNC to be polite! How? Parameters

3. PARAMETERS

Parameters (PARMS) are what are used to tell a TNC how to act. PARMS can tell the TNC to be a very agressive bully, pushing everyone else off the sidewalk, or a very meek individual who always waits for every-one else to go first. PARMS set the personality of the TNC.

Every manufacturer has a slightly different way to "improve" or "redefine" the original "standard" PARMS, so some of what follows may not apply exactly to some TNCs.

PARMS are described here with example values which work for most users. It is these PARMS that must be varied to adapt to your radio, path to the Node, and signal strength as compared to other users for the best results. This list of PARMS is only a list of those which affect other users and is not complete. Only the TNC manual can provide a complete description of the parameters built in to a TNC.

The specific values recommended here are just that, recommendations. If your SYSOP has a different set of recommendations, follow his guidance.

A "handy-dandy" chart of "typical" PARM values is not included here, because there is no set of values that is a panacea. Please review each PARM and make an educated selection of a value for your use as a user.

Times are referred to here in milliseconds or seconds. Many TNCs use different units. For instance, TXDELAY is usually quantified in tens of milliseconds, and to set TXDELAY to 250ms means the "VALUE" of TXDELAY must be set to 25 (25*10 = 250). Only the TNC manual can provide the time units used by the specific TNC.
1000ms = 1 sec.

PERSIST

When the TNC has a packet to send and the frequency is quiet, the value of PERSIST determines the probability of transmitting. This helps the randomness of transmitting thus adding to the chances of not colliding with another user. A value of 255 means you will immediately transmit whenever you have a packet to send and the channel is clear. A value of 64 means you will transmit one time in every four times you ask the PERSIST algorithm for permission to transmit. Some MFJ TNCs implement this algorithm in a slightly different manner and call the parameter "SLOTS" with SLOTS 0 equal to PERSIST 255, SLOTS 128 equal to PERSIST 1.

On a very busy channel, node PERSIST should be 20 or less, and user PERSIST should be 10 - 15. This will make the channel seem "sluggish" when composing mail or sending DX spots, but it will be less agressive and keep you connected. If a Network node is present on the user channel and it carries a substantial portion of the channel traffic, it should run a PERSIST closer to the node value.

Lightly loaded channels can stand higher PERSIST values determined by experimentation.

See also PPERSIST (note two P's in PPERSIST).

SLOTTIME

If the TNC has a packet to send and it has asked PERSIST for permission to transmit and that permission was denied, the TNC waits SLOTTIME time and asks PERSIST again for permission to transmit. Some MFJ TNCs implement this algorithm in a slightly different manner and call the parameter "DEADTIME".

Node SLOTTIME on a busy channel should be 200ms or more, and users should never have the same SLOTTIME as the node. Usually, users attempt to avoid SLOTTIME values in use by other users, and users set SLOTTIME to a much longer time than the node. A good starting place on a busy channel for a user is 250ms.

RESPTIME (T2 Timer)

Acknowledgement packets are delayed at least RESPTIME (response time). If there are other parameters which impose delays, they do not add to RESPTIME. The intent of RESPTIME is to give an acknowledgement packet an opportunity to be "piggybacked" with an outgoing information frame.

If PERSIST is active, RESPTIME can be set to a short time, such as 500ms to 2 seconds. If PERSIST is not active, RESPTIME should be longer than 1 second.

DWAIT

This value was designed to avoid collisions with digipeated packets. A TNC will wait DWAIT time after the radio channel is quiet before sending a packet. If this is less time than RESPTIME, then RESPTIME will apply for acknowledgement packets. If PERSIST is used, DWAIT is not operative or should be set to zero.

If DWAIT must be used, it should be set to one second or more to allow users who have PERSIST to respond first.

TXDELAY

The TNC waits TXDELAY time after keying up the transmitter before sending data. This gives the transmitter time to come on frequency and up to full power before data is transmitted. It also gives the user's receiver time to open squelch and get ready to receive data. TXDELAY should not be appreciably longer than necessary as it uses up channel time. A value of 250ms (.25 seconds) is typical.

FRACK (T1 Timer)

After transmitting a packet requiring acknowledgement, the TNC waits FRACK time before sending the packet again. If the packet is resent, the retry counter is incremented.

If the node sends a DX spot to a user and the node FRACK is set to 15 seconds, the node will wait 15 seconds for an acknowledgement from the user before resending the DX spot.

User FRACK should be set to a LONG time, at least 10 seconds.

CHECK (T3 timer)

Also known as IDLETIME, this timer sets the maximum time for no activity between the two connected stations. When this time is expired, the TNC sends a supervisory frame to see if the other station is still active. If this frame is not responded to in the amount of RETRY attempts, the connection is broken. Check polling often creates a lot of overhead and if the path is marginal to begin with, or conditions change for a few minutes, a disconnect will occur. Generally, all CHECK functions are performed at the node, and the user CHECK time should be disabled. If the TNC does not allow CHECK to be disabled, it should be set to the longest possible time (an hour or more?).

RETRY

If the TNC sends a packet which requires an acknowledgement, such as an information packet or a check packet, and if the TNC does not receive the acknowledgement in FRACK time, the TNC will send the packet again. It will send the packet RETRY times before giving up and performing a "hard disconnect".

PacketCluster nodes on busy channels usually set RETRY to 7 or less, thus disconnecting stations with poor connects in favor of other stations who do not require resent packets.

Users have little call to RETRY to the node, so a value of 10 is typical.

PPERSIST

Some TNCs have this parameter which must be explicitly turned on to enable PERSIST and SLOTTIME. PPERSIST should be ON if available. Note this is PPERSIST (two P's at the beginning), not PERSIST.

ACKPRIOR

Most TNCs do not have this parameter. If ACKPRIORITY is on, then the acknowledgement packets do not have to wait for the PERSIST or SLOTS algorithm to permit transmission. There are two very diverse schools of thought on ACKPRIOR. One says it really helps a busy PacketCluster channel, and the other says it is too agressive and has an adverse affect on channel throughput. The best policy is to leave it off unless requested by the node SYSOP.

OTHER PARMS

There are other PARMS which affect the throughput on a PacketCluster user channel which are not mentioned here. MAXFRAME and PACLEN as set at the node will have an effect, as will user abuse of some PARMS such as CONPERM. This list of PARMS is NOT complete, but covers most of the ground.

Observation:

PARM settings are important when many users share a channel. On a busy channel, users should be very passive to prevent the node from "retrying out" to the user and to prevent the user from "retrying out" to the node. There is no PARM setting by the user that will force the node to send DX spots faster.

Results:

It is difficult to predict how well adjustments of parameters will calm a busy node frequency. One measure is how well the users stay connected during a high activity period, such as a contest. Experience shows that 20 or more users can stay connected to a node even during sustained periods averaging a DX spot every minute or less if the parameters are carefully coordinated. However, one new user can show up in the middle of a contest with aggressive parameters, do a SH/DX 25, and in a short time he is the only station connected!

Conclusion:

Hopefully this discussion will help PacketCluster users to understand the following:

1. What happens whan a node sends a DX spot to a user.
   It takes 1.4 seconds under perfect conditions.

2. What happens when a node sends a DX spot to a lot of users.
   RF riot! Packet pileup!

3. Parameters set TNC personality.
   Passive is better! Passive PARMS maintain a connection.

Prepared by Jay O'Brien, W6GO
w6go@netcom.com

References:

1. DX spot length (75 bytes): personal observation and knowledge, W6GO.
2. Component part measurements, 3% bit stuffing: ARRL Computer Networking conference 4, San Francisco, March 30, 1985, paper by David Engle, KE6ZE, titled "PACKET RADIO TIMING CONSIDERATIONS" which detailed actual times measured by KE6ZE and KA6M at 1200 baud.
3. Discussion of Parameters by Tom Wood, N6IXX: On No Cal/Nev DXPSN.
4. TNC Manuals: Kantronics, AEA, DRSI, MFJ, PacComm, Net/Rom, G8BPQ.
5. Personal knowledge and experimentation: W6GO.