PACGRAM MESSAGING PROTOCOL FOR PACKET NETWORKS

Jay Nugent, WB8TKL
3081 Braeburn Circle
Ann Arbor, Michigan 48108

In the 5th Computer Networking Conference papers David Cheek, WA5MWD, included in his submission an attatchment containing the Pacgram Protocol Definition. In this paper you will find the latest revision of that Pacgram Protocol Definition with a synopsis of Pacgram's creation, where it is at now, and where I hope it will go in the amateur packet radio community.

BEGINNINGS

Pacgram's beginnings date back to the good old days of the BETA board tests. It was recognized early on that we were going to require some standard method of passing Radiogram traffic via packet radio. Though the Radiogram form has been in use for many years, the way in which it is used varies from one network to another and from one mode to another. In this I mean that each network has its own particular way of presenting the contents of the Radiogram over the air for a receiving station to copy.

Each network has developed a method that best suits its needs based on the particularities of the network. Voice traffic nets present the Radiogram contents by reading the Radiogram slowly and in chunks or logical groups, stopping at predefined breakpoints for fills or varification. CW traffic networks have a similar method varying slightly to suit the needs of the CW operators network.

With this in mind it only made sense that the packet radio network, with its totally different conditions, would need to develop its own method of presenting Radiogram traffic. To this brings us the need for a set of objectives to be strived for cr met when designing our new protocol.

OBJECTIVES

0 Provide a 'standard'

The most important of all is to design a messaging protocol that would be the standard that all packet radio traffic would follow. To accomplish this would require that the protocol suit all the needs of the traffic handelers.

0 User readable without software

Since the equipment and capabilities of each packet radio station varies, it would be desirable that the protocol use only printable characters so that traffic handelers that did not have Pacgram software would still be able to operate within the network.

Though a little more difficult when originating traffic, it would be a simple task 'when receiving or forwarding a Pacgram to decode the fields by hand or simply receive the message to disk for later re-transmission.

0 Compatable with other networks

Any traffic handler knows that a piece of traffic can travel through any number of different nets before reaching its destination. These nets may be Packet, RTTY, CW, or Voice traffic networks.

This requires us to keep in mind any limitations that these other networks may have when we are originating and even when forwarding traffic. Pacgram must be designed to filter out any non-printable characters along with a number of the computer keyboard characters that cannot be sent in CW or RTTY modes.

0 Not machine dependent

The Pacgram messaging protocol should not have any machine dependancies. The protocol should not require a specific type of terminal, computer, hardware, or screen format. A session layer protocol can be implemented at each Pacgram station to handle these dependancies. This session layer has been implemented in the current CP/M version of Pacgram to support ADM3 style consoles like the Xerox-820.

- Reduce operator workload

An important gain in utilizing computers is to allow the computer to do as much of the tedious work as possible. When implementing Pacgram we should have the software perform automatic word counting, logging, filing, etc.

(Please refer to David Cheeks paper in the 5th Computer Networking Conference on page 5.115.)

- Simple

The 'KISS' principal seems to apply to everything. An important objective is to keep the Pacgram protocol as simple as possible without giving up important options such as future growth and expansion.

- Forwardable

Since a message forwarding network is already in place we should maintain compatability with it. We should also keep in mind future network designs. Pacgram being simply a string of text characters allows us to upload, download, and make use of the automatic mail forwarding features of the WORLI type of Packet Bulletin Board Systems.

Maybe future PBBS's could contain Pacgram pre and post-proccessors for users that did not have Pacgram software, This would allow them to originate and receive Radiogram traffic while the backbone network utilized the Pacgram protocol.

- Expandable

Pacgram's design should include the ability to be expanded and adopted into other applications. One of these applications may be a software interface that extracts Pacgram fields and inserts them into a database,

As you will see, the latest revision of the Pacgram protocol allows for formtypes other than the ARRL Radiogram so database applications can be extremely flexible.

- Automatic or Unattended

Pacgram implementations have been written that allow for automatic receipt of traffic. This can be extremely handy when the operator cannot be at the terminal at all times such as during an emergency situation. Other implementations could allow automatic printing of all Radiogram traffic to hardcopy as they are received while at the same time saving all other packet activities to disk.

CODING THE PROGRAM

With these objectives in mind I started writing a terminal program that contained the Pacgram messaging protocol. The project started out being written in BASIC since nearly every computer has a BASIC interpreter and that would allow a greater number of machines to support Pacgram. But this failed miserably when it was determined that BASIC simply was not fast enough without being compiled, and this certainly would limit the number of users due to the lack of BASIC compilers for many of the smaller computers.

This lead me to write the Pacgram Terminal Program in 8080 assemoly code to run under CP/M. This would limit the use of Pacgram to only CP/M machines and others that could emulate CP/M such as the IBM-PC, Apple, and some Commodores.

Programming started in February 1984 with software tests both in the Detroit and Dallas areas, followed by many re-writes and updates. Refinements were made to the protocol as problem areas were uncovered.

IMPROVEMENTS

Recent improvements have been to seperate the Area Code into its own field to allow for destination sorting based on Area Codes. The addition of the Formtype field to identify the type of form contained within the Pacgram. This allows us to not only send ARRL Radiogram traffic but Civil Defense traffic as well. Not to mention other formtypes that may be required for special events such as Footraces and Bikethons and databases.

And finally, the addition (but not yet implemented) Application Level Acknowledgement. Since our only acknowledgements are at the Link layer, there was no way of guarenteeing that the Pacgram was properly captured by the receiving station so an additional verification is being considered.

For more details on the Pacgram protocol, see the Pacgram Protocol Definition that follows at the end of this paper.

FUTURE

It is hoped that Pacgram will eventually catch on as the standard for Radiogram, Emergency, and/or Database messaging. The protocol is still developing but appears to be quite dependable and stable at its current revision.

As for the future, good news. In the ARRL FIELD FORUM dated January 1987, after a meeting of the Eastern Area Staff of the National Traffic System in Williamsburg, Virginia, the following resolution was published:

"THEREFORE, the Eastern Area Staff RECOMMENDS that the Eastern Area Staff Packet Committee continue its work towards defining a strategic traffic system which integrates packet, and

FURTHERMORE, the Eastern Area Staff will implement, in the near term, the following National Traffic System usage of packet for a trial period of two years:"

"...and. FURTHERMORE, the Eastern Area Staff recommends the use of the PACGRAM message format, as described by the 5th Networking Conference, during the trial period. "

It is hopeful1 that the NTS Eastern Area Staff will use the most current revision of the Pacgram message format as published below.

[Pacgram Protocol Definition]
(Versions 2.0.3 and later)

By: Jay Nugent WB8TKL
3081 Braeburn Circle
Ann Arbor, Michigan 48108
(313) 9714076

Dated: 860813

PACGRAM is an application software package that runs on the host computer connected to a TNC. The PACGRAM software is responsible for prompting the operator for the proper Radiogram information one field at a time and forms a PACGRAM message from this. The message can then be sent to the TNC for transmission into the amateur packet network.

On the receiving end, PACGRAM decodes the data stream from the TNC for the starting characters of a PACGRAM. When it finds these characters it receives the rest of the message and can later decode it back into the Radiogram format for display on the console, or printed on the printer. Received PACGRAMS are stored in buffer space and/or disk files and may be later retransmitted or forwarded to other stations in the network.

Special characters are used within the PACGRAM to signal the start of the PACGRAM message, the end of the PACGRAM message, and to seperate the fields of information within the PACGRAM message.

These control characters and the protocol are described in the following definition.

----- PACGRAM CONTROL CHARACTER and PROTOCOL DEFINITION

The control characters, and character sequences, used in PACGRAM were based on the unlikelihood that they would ever appear in any part of a normal Radiogram.. Consideration of the CW traffic nets that may handle message traffic generated with PACGRAM was also taken into account since many characters cannot be sent using CW.

For those stations not possesing PACGRAM software, these control characters were selected so that a PACGRAM can be read directly from a terminal and written back into the standard Radiogram form very easily by hand. A Formsmode has been added to the protocol to allow the sending of a directly printable PACGRAM. This enables any station to receive a PACGRAM already formatted to be printed on hardcopy. This form of PACGRAM uses its own starting sequence that can be easily detected by a small computer running BASIC. Once the start sequence is detected, it can then route all output to the printer. The standard end of PACGRAM character is used to indicate the end of the print so that the output can then be routed back to the console.

---- The START of a PACGRAM shall be a pound sign '#' followed by an asterisk '*'.

The purpose of this character sequence is to signal the start of a PACGRAM and differentiate it from any other data sent by the TNC to the host computer. Such as other communications data or commands and responses from the TNC.
The start of the PACGRAM sequence was altered in the second release of this protocol in an effort to avoid false starts caused by WORLI like PBBS's that use the # character in their data.

---- The FORMTYPE sequence shall be 'ARL' for ARRL Radiogram format and shall always be three characters in length followed by the DELINIATION character.

The purpose of this three letter sequence is to identify the form type that is contained within the Pacgram. For the standard ARRL Radiogram form this sequence is set to ARL. Applications requiring message formats other than ARL may define their own field names and order and define their own three letter Formtype.
(This change has been made for releases 2.0.0 and above)

---- The DELINIATION character shall be an asterisk '*'.

This character is present in the PACGRAM to deliniate the Radiogram fields from one another. The absense of data in any one of the fields will cause two consecutive asterisks to appear within the PACGRAM. No filler characters are placed between the asterisks of an empty field.

---- The FIELD ORDER of an ARL PACGRAM is as follows.

NUMBER / PRECEDENCE / HANDELING INSTRUCTIONS / STATION OF ORIGIN CHECK / PLACE OF ORIGIN / TIME FILED / DATE FILED

NAME TO / NUMBER & STREET / CITY / STATE / ZIP / AREA CODE / PHONE NUMBER

TEXT OF THE MESSAGE

SIGNATURE / TITLE OF SIGNEE

Note: The CITY and STATE fields have been seperated into two individual fields. This allows for automated routing based on the destination City and State. Area Code was seperated for the same reason.

---- FIELD LENGTHS and TYPES within the ARL PACGRAM.

The Number field will be limited to 8 characters maximum.
The Check field will be limited to 10 characters maximum.
(This has been expanded to handle ARL type checks)
The Text field is limited to a maximum of 1024 characters.
All other fields are limited to 60 characters.

Fields may contain either alphabetic or numeric characters but those characters that cannot be sent using CW will not be allowed.

Such as the following:
- Pound sign * - Asterisk & - Amperand
% - Percent < - Left arrow > - Right arrow
= - Equals ^ - Carat - Underscore
All control characters including Carriage Return, Linefeed, and Formfeed.
Any non-allowed characters found within a Pacgram will be discarded.

---- The END of a PACGRAM shall be the ampre sign '&'.

The purpose of this character is to signal the end of the PACGRAM.

---- The START of a FORMSMODE PACGRAM shall be '#PAC*' followed by a carriage return.

The purpose of this starting sequence (including the carriage return) is to signal the start of a PACGRAM in the FORMSMODE. A small computer running a simple BASIC program can trap this starting sequence and then direct all its output to a printer.

---- The END of a FORMSMODE PACGRAM shall be an ampre sign '&' followed by a carriage return.

The purpose of the end of FORMSMODE sequence is to signal the end of a FORMSMODE PACGRAM. A small computer running a BASIC program can trap this sequence and then direct its output back to the console.

---- Proposed Application Level Acknowledgement shall be as follows:
#*ACK*Number*Precedence*HX*Station of Origin&

The purpose of the Application Level Acknowledgement is to confirm to the sending station, in a positive manner, that a Pacgram sent by it to another station has in fact been received by the distant station.
Note the Startpac sequence followed by ACK to indicate acknowledgement. The NUMBER, PRECEDENCE, HX, and STATION fields are echoed back by the receiving station for comparison to SENT Pacgrams. A match confirms which SENT Pacgram has been acknowledged. It can then be flagged as ACKnowledeged and logged.
The NUMBER and STATION fields should never be repeated within any operating year so the likelihood of an incorrect ACK is very unlikely.

==========

As a Radiogram has well defined fields in a specific order, so does the PACGRAM. The order in which the fields occur in the Radiogram is the exact order that they will appear in the ARL PACGRAM. For example, here is a sample Radiogram followed by its equivelent data stream in PACGRAM form.

NUMBER: 126 ROUTINE WB8TKL CHECK: 5 ANN ARBOR MI 14302 MAY 21
TO: MIKE NUGENT
123 HOLLYWOOD AVE
HOLLYWOOD, CAL 54321
(818)555-1234
TEXT: HOW IS THE WEATHER X
SIGNED: JAY

And the equivelent in ARL PACGRAM form would be:

#*ARL*126*R**WB8TKL*5*ANN ARBOR MI*1430Z*0521*MIKENUGENT*123 HOLLYWOOD AVE
*HOLLYWOOD*CAL*54321*818*555-1234*HOW ISTHEWEATHERX*JAY*&

The proposed Application Level Acknowledgement to this would be:

#*ACK*126*R**WB8TKL&

As you can see, the length is well within 256 bytes, tne maximum AX.25 packet length. Even with the Paclen set to 256, a single packet could contain a fairly large text field. Shorter Paclens can be used as network conditions require. The Pacgram will simply be re-assembled back into its original form frame by frame.

You can see the benefits of using PACGRAMs over the voice or CW methods of sending traffic. This packet can be sent in just a fraction over one second at 1200 bps, an enormous improvement over existing amateur traffic systems. Pacgram also prompts the operator for the fields in the correct order, so fewer mistakes are made.

Also notice in my example, that since I left the fields for Handling Instructions and Title blank, that PACGRAM simply put no data between the two asterisks. This is neccessary to maintain the field count for decoding of the PACGRAM at the receiving end and also wastes as little of the transmission bandwidth as possible.

Happy Packet ing -WB8TKL