

PACSAT Protocol Suite - An overview

Harold E. Price, NK6K
Jeff Ward, G0/K8KA

ABSTRACT

A Low Earth Orbiting “**Pacsat**” has been described in the past as an orbiting bulletin board system. This is an over-simplification. A PACSAT is a multi-channel, full duplex device, with **short**, periodic **access times dictated** by orbital mechanics. These attributes mandate a different approach than the standard command-line interpreter style of BBS if the full potential of a PACSAT is to be realized.

The authors propose a new methodology for a PACSAT, and have developed several new protocols to implement more efficient access. These protocols all use AX.25, either in connected mode or with UI frames. This paper provides a description of the access model, and an overview of the new protocols.

Background

The authors have been struggling with the question “How can we make the best use of a bandwidth-limited low earth orbiting digital **store-and-forward** system with a worldwide, unstructured? heterogeneous user base” since an amateur Packet Radio satellite was first discussed in 1982. We began on air experimentation with the UoSAT-2 (UO-11) Digital Communications Experiment in December, 1984. In the following five and one half years, we’ve looked at where a resource like a PACSAT best fits in to the network as a whole. As a result of our study, we are proposing the use of a broadcast protocol as the basic **downlink method**, and a “file server” rather than a BBS application as the basic service offered. This document provides a brief overview of these conclusions, the companion specification documents provide the implementation details.

This paper and the companion protocol specification papers assume that the reader has a basic understanding of the current packet radio satellites, for additional background. see references [1] through (6).

PACSAT

PACSAT is generic term in the amateur radio service for a low earth orbiting spacecraft which carries

a large on-board memory for the purpose of data storage and retrieval by groundstations. A PACSAT can be the entire mission of a spacecraft, such as **AMSAT-NA’s** AO-16, or a minor adjunct, such as the DCE on UO-11. The paper refers to the current “PACSAT” spacecraft - the University of Surrey’s UoSAT-3 (UO-14) and the **AMSAT Microsats** AO-16 and LO-19. These spacecraft will be the hosts of software developed by the authors which implements the protocols described herein.

Each of these spacecraft are different. AO-16 and LO-19 are the most closely related, based on **AMSAT’s** **Microsat** design. From the **user’s** point of view, they have four 1200 bps **uplinks** and one 1200 bps **downlink**. These are switchable to 4800 bps, but no ground modems exist at this time. UO-14 has a single **uplink** and **downlink**, at 9600 bps. Although the **onboard** computers are different, they are compatible at the application software level, permitting the same software to be used on all three.

In spite of these differences? all of these spacecraft share the following attribute: each is a bandwidth limited device. The number of **uplinks** and **downlinks** is much less **than** the number of users, and the capacity of the link is much less than the offered load. Each is only visible to a particular user for about **14** minutes, four or five times a **day** at middle latitudes. We feel that this is the critical

design driver, and the access methods must be **optimized** with this in view.

Keep in mind, however, that even while subject to access time limitations, the satellites can still move a prodigious amount of data, especially when compared to the current amateur radio long haul network standards. A typical gateway station, moving traffic from the US to the UK on 20MHz at 300 baud, assuming the band is open for 16 hours, could move 1.7 million bytes of data per day, if the link was 100% efficient. The average HF link is not 100% efficient, at best it is perhaps 30% efficient. The link is only half duplex, so this data transfer is only way only.

UO-14, even with only 56 minutes of access time per day at 9600 baud, can move 3.2 million bytes of data in one direction. The excellent link quality of the current PACSATs, combined with their full duplex nature and the protocols we are proposing, can approach 90% efficiency. Full duplex means transfers can occur in both directions simultaneously, so that UO-14 could move nearly 5.7 million bytes of data between the US and the UK in a 24 hour period, vs. .5 million bytes over an HF circuit.

The desire to realize this potential is the reason we choose some non-traditional (for the amateur radio service) access methods for PACSAT. These methods, broadcasting and file server, are discussed below.

Broadcasting

A spacecraft is inherently a broadcast device. It transmits from on high, and many users can hear it at the same time.

To optimize the available **downlink** time, we are recommending the use of a broadcast protocol. This protocol adds information to the basic Ax.25 data frame to permit many stations to make simultaneous use of a single file download session. When one station in Maryland requests the current orbital element sets, there is no need for stations in Toronto and Miami to do the same, they should be able to make use of the information as it is downlinked to Maryland if they are all in view of the satellite at the same time. To make use of a broadcasted frame of data, each frame must be tagged with the file it belongs to and the position within that file that the data belongs in.

There should also be enough information for a station to determine if it has all of the data belonging to a file, and if not, to request that just the missing parts of the file be retransmitted. The specification titled "PACSAT Broadcast Protocol" describes a method of providing this additional information.

With a broadcast protocol, a groundstation can simply monitor the **downlink** and accumulate files of data. Since files gathered in this way will have been unsolicited, the format of the contents may not be known to the user. For example, if one asked for a file of NASA format orbital elements, one can make a good guess that the resulting file contains NASA format orbital elements. However, if a "random" file is captured, its contents may not be understandable simply from inspection. Some additional information, such as a file name, data type, description, creation date, etc., may be required. Each broadcasted file? therefore, needs a header in a standard format with this information. The specification titled "PACSAT File Header Definition" describes a method of providing this information.

We hope that the broadcast protocol promote efficient use of the **downlink**. It should reduce the number of requests for files of general interest. It should also reduce the **uplink** loading. since a broadcasted file does not receive an ack for each frame or group of frames. In the best case, only one "ack" is sent for an entire file, and that would be the request to stop broadcasting it.

Even **though** the sky-to-ground link is broadcast in nature, the ground-to-sky link is not. PACSAT "sees" many ground stations at one time. For this reason? a 'connected-mode, non broadcast file transfer method is also defined. and is described in the paper on "PACSAT File Transfer Level 0".

File Server

As a data transfer and storage device. a PACSAT can serve a multitude of purposes. It can store telemetry, digitized voice and video images, personal mail, forwarded mail, or anything else the can be stored in a computer file. Mail forwarding is a good example of an excellent use of a PACSAT. AO-16's 1200 baud link could easily be **used** to transfer 240k bytes of uncompressed forwarded mail in each direction between California and England in 24 hours, with just one morning

and evening pass over each location. UO-14's 9600 baud link could move 1.6 Mb of data in the Same time. A PACSAT can store up to 8Mb of data. This would make a powerful addition to the current HF relay network.

The problem, however, is that the current amateur network is in a state of flux. New addressing schemes are proposed every few weeks, new routes and new ways of routing are proposed, tried, discarded or modified. This is good. Implementing the software on a spacecraft to follow these shifting designs is difficult, however. The testing required for the spacecraft is more rigorous, especially on the Microsats, where the same computer is used for the BBS and to keep the batteries charged. Faulty forwarding code could crash the computer, which could cause damage to the batteries or reduce their life expectancy.

The amount of program memory is limited on the spacecraft as well. To counter the effects of high energy particles above the earth's atmosphere which cause memory bits to be changed, the PACSATs use 12 bits to store 8 bits of program data. The extra bits are used to correct for single bit errors. To keep the cost down, and to reduce the power used (AO-16's CPU uses about 500 milliwatts, on average), only 256k bytes of program space is available. (This should not be confused with the message storage space, which is much larger than the program memory, and is protected with a software algorithm using three 3 bytes to protect 253 bytes of data. Because this memory is protected with software, it is not suitable for storing a running program, since a program can not protect its executing instruction.)

We have a desire, then, to keep the spacecraft code simple and stable, while still allowing it to be a useful part of the changing amateur network.

We propose that the spacecraft be primarily used as a file server, moving data files from one point to another. The PACSAT would have no knowledge of the contents of the files, nor would it take an active role in the forwarding of mail messages. Groundbased software could, however, make the PACSAT system look like a familiar BBS to the user, and it could intelligently forward mail.

A PACSAT will know how to receive and transmit a standard file format. All files will have a standard header, the same one that is used by the broadcast protocol. It will also know how to select

files for transmission based on the contents of the header. This feature can then be used by ground-station software to emulate any desired user interface.

For example, assume that a user wanted to send a personal mail message to a friend. In the current terrestrial environment, he would connect to a BBS, which would lead him in a question and answer session something like this:

<u>Remote Computer</u>	<u>User</u>
What do you want?	Send message
To whom?	Fred
Title?	Club meeting
Message?	Meeting at 8 p.m.
What do you want?	Read new mail.
Message #200	
.....	

Using the PACSAT system, exactly the same exchange would take place, except that the conversation is between the user and his local computer. The message is stored for later transmission to a PACSAT. The read new mail request is also stored. The next time the PACSAT comes overhead, the computer does the following:

- 1) builds a file with a standard PACSAT header. The header says that the file contains a mail message, from you, to Fred.
- 2) The file is compressed: and sent to PACSAT.
- 3) The local computer then sends a message to PACSAT that says "send the next file who's header meets the following criteria: it's a mail message type. the destination is me. and the file number is bigger than x".

"x" is the number of the last file received on the ground, and is kept by the local computer. After the pass, the local computer can now print any new mail received. To the user, it looked pretty much the same.

What about file forwarding? A gateway would need to know what type of mail it could forward. Let's assume that the routing scheme of the week is based on a hierarchical string containing states. like nk6k.ca.usa, and this gateway handles mail to CA, NV, and OR. The gateway would send a message to PACSAT containing the following request:

“Send the next file who’s header meets the following criteria: it’s a forwarded message, and the destination string contains ‘?.ca.’ or ‘?.nv.’ or ‘?.or.’, and the download count is 0.”

The file would be received, decompressed, and imported into the standard BBS program after the pass.

In this way, the ground program can be as simple or as complex as required, the PACSAT only needs to know how to select a file for transmission based on the contents of fields in the standard file header.

Summary

These two ideas, broadcasting and file server, are certainly different than the current common usage of packet radio on the amateur bands. **We feel** that this is the best approach for the special case of a PACSAT, however, and that with suitable groundstation software, these concepts can be integrated into the mainstream.

Implementation Status

Prototype implementations of **all** of the protocols discussed in this group of papers are running on UO-14 as of late July, they should be running on the Microsats by the time of the ARRL conference. Prototype ground software is also running. We plan to make the source code for simple versions of the ground portion of the system available **asap**. Executable versions for the IBM PC will be made available as shareware, with the proceeds going to AMSAT-UK and AMSAT-NA to further **development** of future **PACSATs**. Fully integrated, automated, color graphic, “**all** singing and dancing” software will be available for sale by **AMSAT-UK** and **AMSAT-NA** later in the year. Like **QUICKTRAK** and **InstantTrak**, the proceeds from this commercial quality software will go to finance future amateur satellite endeavours.

We hope that other software authors will use the documentation and source to develop support for non IBM **PC** systems. The contents of these papers are sufficient to allow programmers to begin implementing their own software now.

Correspondence

The authors may be reached at:

Telemail: **HPRICE** or **UOSAT**
CompuServe: **71635,1174**
Packet: **NK6K @ WB6YMH** or
G0K8KA @ GB2UP
Internet: 71635.1174
@COMPUSERVE.COM
Mail: Jeff Ward
UoSAT Unit
University of Surrey
Guildford, Surrey **GU2 5XH**
UK

REFERENCES

- [1] Loughmiller D., and McGwier. R., “Microsat: The next Generation of OSCAR Satellites - Part 1”, *QST*, May 1989, pp. 37-40.
- [2] Loughmiller D., and McGwier. R., “Microsat: The next Generation of OSCAR Satellites - Part 2”, *QST*, June 1989, pp. 53-54,70.
- [3] Clark, T., “Amsat’s Microsat/Pacsat Program”, *ARRL 7th Computer Networking Conference?* pp. 41-47, Columbia, Maryland, 1 October 1988.
- [4] Ward, J., “The UoSAT-D Packet Communications Experiment”, *ARRL 7th Computer Networking Conference*, pp. 186-193, Columbia, Maryland, 1 October 1988.
- [5] Price, H., and McGwier, R., “Pacsat Software”, *ARRL 7th Computer Networking Conference*, pp. 145-149, Columbia, Maryland, 1 October 1988.
- [6] Johnson, L., and Green, C., “Microsat Project - Flight CPU hardware”, *ARRL 7th Computer Networking Conference*, pp. 186-193, Columbia, Maryland, 1 October 1988.