Communications Infrastructure for the MOST Microsatellite Project (excerpt)

Laura Halliday

Chapter 3

File transfer with PB and FTL0

3.1 Introduction

The unique file transfer protocols used by microsatellites are an attempt to make optimal use of an expensive resource: the satellite downlink.

The first digital amateur satellites used an adaptation of a terrestrial file transfer protocol. The key adaptation was addition of facilities to permit files to be transferred—in either direction—over several passes of the satellite. While this worked, it only allowed one ground station to access the satellite at one time. Other ground stations could hear the downlink transmissions from the satellite, but could not make use of them.

The solution was a *broadcast protocol*. The assumption, based on experience with early digital satellites and terrestrial networks, was that files were of general interest. By broadcasting a single copy of a file to all interested ground stations, system throughput and efficiency could be drastically increased.

The broadcast protocol may be characterized as a selective request repeat protocol. This is a protocol where ground stations initiate a file transfer, and then request retransmission of needed portions until the file is complete. The broadcast protocol also permits receive-only operation, where a passive station may receive files by merely listening to broadcasts initiated by others.

The satellite environment facilitates using different frequencies, in different amateur radio bands, for communications between ground stations and the satellites. This eliminates channel contention,

a major consumer of channel capacity in terrestrial networks: terrestrial stations that wish to transfer files must share the channel with the files themselves.

This chapter will discuss the broadcast protocols in detail, examining the interaction between ground stations and satellites.

3.2 Terminology

The **P**ACSAT **B**roadcast Protocol, PB, is the protocol used for downloads on current satellites. **F**ile **T**ransfer **L**evel **0**, FTL0, was once the protocol for uploads and downloads, but is now only used for uploads.

In all discussions of the PACSAT broadcast protocol times are stored as 32 bit Unix times, the number of seconds since 0000 UTC 1 January 1970. All multi-byte quantities are stored least-significant byte first, "little-endian" byte order, as used by Intel microprocessors.

3.3 Downloading files with PB

The **P**ACSAT **B**roadcast protocol is the protocol currently in use for file transfer on the amateur satellites listed in Table 3.1 (1).

Table 3.1: Amateur satellites using PB

Satellite	**Downlink**
AO–16	1200 baud BPSK
UO–22	9600 baud MSK
KO–23	9600 baud MSK
KO–25	9600 baud MSK
TO–31	9600 baud MSK
UO–36	9600 and 38400 baud MSK

3.3.1 PACSAT File system

In order to support a large number of users with potentially disparate computer systems, the satellites use their own *PACSAT file system*. This is a very simple flat numbered file system. There is a single directory and the files are identified by a serial number, assigned by the file server at upload. This has proven adequate in experience, and is simple to implement with the limited computer resources available on the satellites. By adding additional information to files (PACSAT file headers, Section 3.3.2) it is possible for ground stations to select files of interest for download.

3.3.2 PACSAT file headers

To facilitate processing by both the satellite and ground stations, files have additional information added at creation time: the *PACSAT file header*.

The PACSAT file header provides information about the file: when it was uploaded, by whom, the type of data it contains, when it should be automatically deleted, and so forth. The header information is stored as a series of keys and values at the beginning of the file, before the user information. When a file is uploaded this information is added by the uploading software. When the file is downloaded this information is removed before making the file available to the user.

The PACSAT file header begins with a special marker `0xaa55`. It ends with an item whose header ID and length are 0, producing the bit pattern `0x000000`. In all header items the identification code is a 16 bit integer, and the length is an 8 bit unsigned integer. The header items listed in Table 3.2 are mandatory (2).

A sample PACSAT file header was obtained from AO–16, illustrated in Figure 3.1 and decoded in Table 3.3. The file is a satellite callsign log file generated on 8 December 1999. Note that this PACSAT file header does not conform to the PACSAT file header documentation (3), which states that header items must be provided in ascending numeric order. The position of item `0x12`, Upload Time (originally optional, now mandatory), violates this aspect of the standard.

Table 3.2: Mandatory PACSAT file header items

Code	Length	Description
0x01	4	File number
0x02	8	Text file name
0x03	3	Text file extension
0x04	4	File size in bytes
0x05	4	File creation time
0x06	4	Last modification time
0x07	1	Set if a **S**ingle-**E**vent **U**pset has occurred in the file
0x08	1	Type of data stored in the file
0x09	2	Body checksum: 16-bit XOR of all bytes in the file body
0x0a	2	Header checksum: 16-bit XOR of all bytes in the file header
0x0b	2	Location of user data in the file

```
aa 55 01 00 04 e0 21 00 00 02 00 08 43 4c 39 39
31 32 30 38 03 00 03 20 20 20 04 00 04 4e 0b 00
00 05 00 04 d5 b0 4d 38 06 00 04 d6 b0 4d 38 12
00 04 d6 b0 4d 38 07 00 01 00 08 00 01 d9 09 00
02 d4 99 0a 00 02 b6 0d 0b 00 02 50 00 00 00 00
23 30 54 4d 53 41 54 2d 31 00 20 20 20 20 50 59
33 50 51 00 00 4b 42 32 57 51 4d 00 56 45 32 4c
41 00 00 56 41 33 53 46 4c 00 4e 33 45 56 51 00
00 4b 38 54 4c 00 00 00 4e 56 38 46 00 00 00 4a
4a 33 59 55 4a 00 56 4b 33 4b 4f 53 00 56 4b 33
4a 44 47 00 4c 57 31 44 58 50 00 58 51 35 42 52
43 00 43 45 33 53 53 42 00 43 45 35 4e 47 00 00
41 44 34 45 42 00 00 4b 45 34 5a 58 57 00 4b 45
34 4b 4f 4c 00 4b 4e 34 57 5a 00 00 56 45 33 46
52 48 00 4e 38 58 4b 5a 00 00 56 45 33 42 43 47
00 57 42 38 48 52 4f 00 57 30 53 4c 00 00 00 4b
```

Figure 3.1: Sample PACSAT file header

Table 3.3: Decoded PACSAT file header

Raw value	**Description**	**Engineering units**
`0xaa55`	Header marker	
`0x0001 04 000021e0`	File number	
`0x0002 08 434c393931323038`	File name	“CL991208”
`0x0003 03 202020`	File extension	“ ”
`0x0004 04 00000be4`	File size	3028 bytes
`0x0005 04 384db0d5`	Creation time	031357 UTC 8 December 1999
`0x0006 04 384db0d6`	Last modification time	031358 UTC 8 December 1999
`0x0012 04 384db0d5`	Upload time	031358 UTC 8 December 1999
`0x0007 01 00`	SEU flag	None detected
`0x0008 01 d9`	File type	217 (callsign log)
`0x0009 02 99d4`	Body checksum	
`0x000a 02 0db6`	Header checksum	
`0x000b 02 0050`	Body offset	80 bytes
`0x000000`	End marker	

Table 3.4: PACSAT broadcast protocol frame header

Item	Length	Description
Flags	1	See Table 3.5.
File number	4	PACSAT file number
Offset	4	Location of first information byte
Time old	4	See text
Time new	4	See text

3.3.3 Broadcasts

The basic unit of communication between the satellite and ground stations is the *broadcast frame*. The satellite broadcasts to all ground stations within its footprint, and stations may use or ignore what they hear at their discretion. The information the satellite broadcasts is in response to requests transmitted by ground stations. The satellite broadcasts several kinds of information, which will be examined in turn.

3.3.4 Directory broadcasts

The first kind of broadcast from the satellite is a *directory broadcast.* This informs ground stations what files are available for download. A directory broadcast (4) consists of a frame header followed by a PACSAT file header (Section 3.3.2). It is transmitted as an AX.25 UI frame (Section 2.2) with a protocol id of `0xbd` and a destination address of `QST-1`.[1]

The frame header contains the information described in Table 3.4. The *flags* field contains several bit fields, as described in Table 3.5. The *time old* and *time new* fields identify possible different versions of the file. Given the upload time for the file, there are no other files other than the one specified by this particular file number with *time old* $\leq$ *upload time* $\leq$ *time new*.

The *offset* field may not have been implemented in the implementations studied: this value was zero in every directory broadcast received.

A sample directory broadcast was recorded from AO–16. The raw data are presented in Figure 3.2, and are decoded in Table 3.6. After the header information we note the `0xaa55` bit pattern denoting the beginning of a PACSAT file header.

Table 3.5: PACSAT broadcast *flags* field

Bits	Description
0–1	Frame type. `00` is a PFH broadcast frame. All others reserved.
2–3	Version identifier. Currently always `00`.
4	Always zero: server generated frame.
5	Set if the last byte of the frame is the last byte of the PFH.
6	Set if this is the newest file on the server.
7	Reserved.

```
20 67 AE 00 00 00 00 00 00 E0 7E 3C 38 EE CF 3D
38 AA 55 01 00 04 67 AE 00 00 02 00 08 42 4C 39
39 31 31 32 34 03 00 03 20 20 20 04 00 04 E0 06
00 00 05 00 04 49 2D 3B 38 06 00 04 E1 7E 3C 38
12 00 04 E0 7E 3C 38 07 00 01 00 08 00 01 CA 09
00 02 FB 44 0A 00 02 84 0C 0B 00 02 50 00 00 00
00 9D 3D
```

Figure 3.2: Sample PACSAT directory broadcast

Table 3.6: Decoded PACSAT directory broadcast

Item	Raw value	Engineering units
Flags	`0x20`	See Table 3.7
File number	`0x0000ae67`	
Offset	`0x00000000`	Not implemented?
Time old	`0x383c7ee0`	001216 UTC 25 November 1999
Time new	`0x383dcfee`	001022 UTC 26 November 1999

Table 3.7: Decoded PACSAT broadcast flags field

Bits	Description	Engineering units
00	Frame type	PFH broadcast frame
00	Version identifier	0
0	Frame origin	Server generated frame
1	Last byte in PFH?	Yes
0	Newest file on server?	No
0	Reserved	

The *flags* value in Table 3.6 decodes to the information shown in Table 3.7.

3.3.5 File broadcasts

When a ground station requests a file for download it is transmitted in a series of *file broadcasts.* These consist of a series of pieces of the file, with control information so that ground stations may reassemble them. Like all broadcasts in the protocol, these transmissions may be received by any interested ground station—even a passive one that makes no requests at all.

File broadcasts are addressed to `QST-1`, and use AX.25 protocol `0xbb`.

A file broadcast packet has minimal control information, followed by the file data. The elements of a file broadcast packet are listed in Table 3.8. A number of options are defined for the Flags field, but the only one used in present implementations is the bit specifying the piece location within the file by byte count, rather than by block count. The Flags field was thus always `0x02` in all received packets.

Sample packets were obtained from AO–16 broadcasts. Figure 3.3 illustrates the broadcast of the first piece of a file, and it is decoded in Table 3.9. Note that the first bytes following the broadcast header are `0xaa55`, the PACSAT file header. Figure 3.4 illustrates the broadcast of a piece from the interior of a file, and it is decoded in Table 3.10.

[1] *QST* is a signal used in the amateur radio service for information of interest to all radio amateurs.

Table 3.8: Data elements in a file broadcast packet

Item	Length	Description
Flags	1	Bit field. Always `0x02`.
File id	4	PACSAT file number
File type	1	PACSAT file type
Offset	2	Offset of this piece in the file: low-order 16 bits
Offset MSB	1	High-order 8 bits of offset

```
02 7E AE 00 00 C9 00 00 00 AA 55 01 00 04 7E AE
00 00 02 00 08 41 4C 39 39 31 31 32 39 03 00 03
20 20 20 04 00 04 C1 03 00 00 05 00 04 64 C5 41
38 06 00 04 5B FC 41 38 12 00 04 5A FC 41 38 07
00 01 00 08 00 01 C9 09 00 02 EB AD 0A 00 02 88
0D 0B 00 02 50 00 00 00 00 12 19 5F C5 41 38 00
00 00 A0 67 00 00 31 00 00 00 1C 17 00 00 EE 9D
29 2F 12 19 6E C5 41 38 00 00 00 CF 9F 00 00 30
00 00 00 1B 17 00 00 C2 C0 D3 34 12 19 72 C5 41
38 00 00 00 A6 A1 00 00 30 00 00 00 1B 17 00 00
04 AB 2B 35 12 19 75 C5 41 38 00 00 00 CF A2 00
00 30 00 00 00 1B 17 00 00 69 6F 63 35 12 19 79
C5 41 38 00 00 00 68 A6 00 00 30 00 00 00 1B 17
00 00 C3 B4 37 36 11 19 7F C5 41 38 00 00 00 5A
AB 00 00 30 00 00 00 1B 17 00 00 B6 02 7C 37 11
19 84 C5 41 38 00 00 00 5B AB 00 00 31 EE 27
```

Figure 3.3: Sample PACSAT file broadcast: beginning of file

Table 3.9: Decoded file broadcast packet: beginning of file

Item	Raw data	Engineering units
Flags	`0x02`	
File id	`0x0000ae7e`	
File type	`0xc9`	201 (activity log)
Offset	`0x0000`	File offset 0
Offset MSB	`0x00`	(first piece)

```
02 7E AE 00 00 C9 E8 01 00 19 FF DF 41 38 C1 8E
00 8A AC 00 00 00 00 00 0C 00 00 00 62 0A 00
00 04 15 05 E0 41 38 C1 8E 03 04 00 00 00 00 00
00 00 00 00 00 00 03 10 1B E0 41 38 C2 8E 03 43
58 36 44 44 00 00 0C 19 22 E0 41 38 C2 8E 00 8A
AC 00 00 00 00 00 00 0C 00 00 00 62 0A 00 00 04
15 23 E0 41 38 C2 8E 03 04 00 00 00 00 00 00 00
00 00 00 00 03 10 3C E0 41 38 C3 8E 03 43 58 36
44 44 00 00 0C 19 59 E0 41 38 C3 8E 00 8A AC 00
00 00 00 00 00 0C 00 00 00 62 0A 00 00 04 15 5C
E0 41 38 C3 8E 03 04 00 00 00 00 00 00 00 00 00
00 00 03 10 77 E0 41 38 C4 8E 03 43 58 36 44 44
00 00 0C 19 78 E0 41 38 C4 8E 00 8A AC 00 00 00
00 00 00 0C 00 00 00 62 0A 00 00 04 15 81 E0 41
38 C4 8E 03 04 00 00 00 00 00 00 00 00 00 00 00
09 11 7B E4 41 38 00 00 00 2E 17 00 00 9C 48
```

Figure 3.4: Sample PACSAT file broadcast: file interior

Table 3.10: Decoded file broadcast packet: file interior

Item	Raw data	Engineering units
Flags	`0x02`	
File id	`0x0000ae7e`	
File type	`0xc9`	201 (activity log)
Offset	`0x01e8`	File offset 488
Offset MSB	`0x00`	

Table 3.11: Destination callsign usage in PB

Callsign	**Meaning**
BBSTAT	General status information.
PBLIST	Unlisted calls are invited to make requests.
PBFULL	Queue full: do not transmit.
PBSHUT	Satellite closed.

3.3.6 Status transmissions

Satellites broadcast status information so that ground stations may decide when to download files. The content and meaning of status messages is documented as part of the broadcast protocol.

Status broadcasts advise if the satellite is available for file transfers. They advise if the broadcast queue is full or empty, and if it is not full, who is in it. They also advise who is using the satellite uplink frequencies, particularly important if the satellite has more than one uplink receiver. All these transmissions are AX.25 UI frames, addressed to different destination callsigns for different broadcast functions, as summarized in Table 3.11. The status broadcasts are human-readable, but are typically interpreted by groundstation software.

The first status message a ground station must receive is the satellite status. This advises that the satellite is open for business:

```
Open ABCD:
```

The code ABCD shows that the satellite[2] has four uplink frequencies, and that none of them are in use. To assist ground stations in selecting uplink frequencies, the satellite advises which frequency a ground station is using for FTL0 file transfers:

```
Open A CD: N0ALJ
```

Once the Open message has been received a ground station may transmit a request to the satellite. The satellite acknowledges the request with the only non-broadcast message in the protocol:

[2]The example is based on data received from AO–16.

```
OK VA3SFL
```

If the satellite cannot fulfil the request it issues an error message instead:

```
NO -2 VA3SFL
```

A common error is a request for a non-existent file. The broadcast queue may be full.

While file transfers are in progress the satellite advises ground stations of the broadcasts it has prepared. The format is a list of the callsigns for which a broadcast is queued. The `\D` notation indicates a directory broadcast for the station:

```
PB: N0ALJ N5ZNL VA3SFL\D W4SM\D
PB: W4SM N5ZNL VA3SFL
```

If the broadcast queue is empty the satellite advises ground stations of the fact:

```
PB: Empty
```

The satellite may be closed for uploads:

```
Shut: ABCD
```

An additional useful transmission is the `B:` message:

```
B: 1691718618
```

This is a running count of bytes transmitted by the spacecraft file server. While it is not possible to evaluate bit error rate under AX.25, it is still possible to compare the values in the `B:` message to the byte count received by the ground station and thus evaluate link quality. WiSP presents this information as a percentage efficiency.

Table 3.12: File fill request format

Component	Item	Size
Header	flags	1 byte
	file id	4 bytes
	block size	2 bytes
Hole list	offset	2 bytes
(repeated as necessary)	offset msb	1 byte
	length	2 bytes

Table 3.13: File fill request *flags* format

Bit number	Item
0–1	`00`: start sending file
	`01`: frame contains a hole list
2–3	Version, currently `00`
4	Reserved, must be `1`
5–7	Reserved, must be `000`

3.3.7 Fill requests

Ground stations transmit *fill requests* to advise the satellite of their wishes. A ground station may request a file fill or a directory fill. The formats for both requests are the same.

A station that wishes to make a request of the server transmits the request as an AX.25 UI frame, using protocol id `0xbb` for file requests and `0xbd` for directory requests, the same as the broadcast messages from the satellite. The format of a file fill request is illustrated in Table 3.12. The format of a directory fill request is illustrated in Table 3.14. Note the similarity of the two formats.

The block size parameter is fixed at 244 bytes, resulting in a total AX.25 frame length of 255 bytes, the maximum length that current satellite AX.25 implementations can handle. The hole list is the gaps in the file that the ground station wishes to fill.

The *flags* field in a file fill request frame is defined in Table 3.13.

Table 3.14: Directory fill request format

Component	Item	Size
Header	flags	1 byte
	file id	4 bytes
	block size	2 bytes
Hole list	start time	4 bytes
(repeated as necessary)	end time	4 bytes

Table 3.15: Directory fill request flags format

Bit number	Item
0–1	`00`: Directory fill request
2–3	Version, currently `00`
4	always `1` indicating a client-generated frame
5–7	Reserved, must be `000`

In the case of a directory fill the “holes” are time intervals in which the ground station has no information on files.

The *flags* field in a directory fill request frame is defined in Table 3.15.

3.3.8 An example download sequence

To illustrate how the different components of the PACSAT broadcast protocol interact, here is a sample download sequence, from initial acquisition of the satellite to the file being complete at the ground station.

In this sample the ground station uses the callsign `VA3SFL`, while the satellite callsign is `MSTSAT`. The satellite broadcast callsign is by convention `MSTSAT-11`, while the BBS (**B**ulletin **B**oard **S**ystem—a holdover from early satellites) callsign is then `MSTSAT-12`. By convention, the ground station SSID is 0, resulting in `VA3SFL-0` as the complete ground station callsign.

The ground station is initially idle. It waits for a status indication from the satellite.

The ground station receives a message:

From `MSTSAT-12` to `BBSTAT-0`, "`Open:  AB:`"

The satellite is now available, and the ground station will now typically request a directory fill. The satellite acknowledges the request:

From `MSTSAT-11` to `VA3SFL-0`, "`OK VA3SFL`".

The file server adds a directory broadcast to the broadcast queue, and adds the information to the broadcast message:

From `MSTSAT-11` to `PBLIST-0`, "`PB: VA3SFL\D`".

The PB message is likely to contain the callsigns of other stations accessing the satellite.

The ground station receives the directory update, applies download rules (or acts on human instructions) and starts to download file `A123`. It issues a request for the satellite to start transmitting the file, which causes the satellite to issue an acknowledgement:

From `MSTSAT-11` to `VA3SFL-0`, "`OK: VA3SFL`".

The satellite then queues the broadcast of the first piece of the file and updates the broadcast message:

From `MSTSAT-11` to `PBLIST-0`, "`PB: VA3SFL`".

Again,other stations may be in the broadcast queue. The satellite will proceed to broadcast information queued in response to requests by other ground stations. The ground station will receive these transmissions and add them to its own files.

When the broadcast request is at the head of the queue the satellite transmits a piece of the file as several frames, each from `MSTSAT-11` to `QST-0`. Other ground stations in the satellite's footprint receive the pieces of the file.

The ground station notes from the PACSAT file header that the file is of length 2048 bytes and reserves storage accordingly. After its turn in the broadcast queue has expired it notes that it has received bytes 1 to 512 and 1024 to 1512. The ground station issues a fill request for file `A123`, bytes 513 to 1023, and 1513 to 2048. The satellite doesn't hear the request due to another station

issuing a request at the same time. The ground station times out and repeats the request. The satellite responds:

> From `MSTSAT-11` to `VA3SFL-0`, "`OK: VA3SFL`".

The satellite then adds a broadcast of the pieces of file `A123` to the broadcast queue, and updates the broadcast queue message:

> From `MSTSAT-11` to `PBLIST-0`, "`PB: VA3SFL`".

When the broadcast reaches the head of the queue the satellite broadcasts the requested pieces. The ground station notes that the file is now complete, saves it for further processing, and moves on to the next file of interest.

Note that this exchange is public. A receive-only station could have learned of the file and received it by merely listening to the messages from `MSTSAT-11` to `QST-0`.

3.4 Uploading files with FTL0

While FTL0 is obsolete for downloading files, it is still the current protocol for uploading files. This section will thus only discuss the use of FTL0 for uploads.

Unlike PB, which uses AX.25 UI packets, FTL0 uses AX.25 in connected mode. There are thus procedures for logging in, logging out, and the notion of a session. FTL0 itself is simple, but it uses the facilities of a complex underlying protocol. This is conceptually similar to the FTP protocol in the TCP/IP protocol suite.

3.4.1 Packet format

FTL0 defines a standard format for packets. They consist of a command code, a data length, and data, as applicable, in the format described in Tables 3.16 and 3.17. The original FTL0 specification documents packet types relating to downloading files, but these are obsolete and are not documented here.

3.6 Notes and references

1. Source: AMSAT News Service. List current as of December 1999. UO–36 was still undergoing testing and on-orbit checkout.
   Internet: `http://www.amsat.org/amsat/news/ans.html`

2. J. Ward and H. Price, “PACSAT File Header Definition”, paper presented at ARRL 9th Computer Networking Conference 1990.
   Internet: `ftp://ftp.amsat.org/amsat/satinfo/pacsat/pacdoc.zip`

3. J. Ward and H. Price, “Regular Broadcasting of PACSAT File Headers”, presented at the ARRL 9th Computer Networking Conference 1990. Despite the title, this is the documentation for the broadcast protocol in its present form.
   Internet: `ftp://ftp.amsat.org/amsat/satinfo/pacsat/pacdoc.zip`