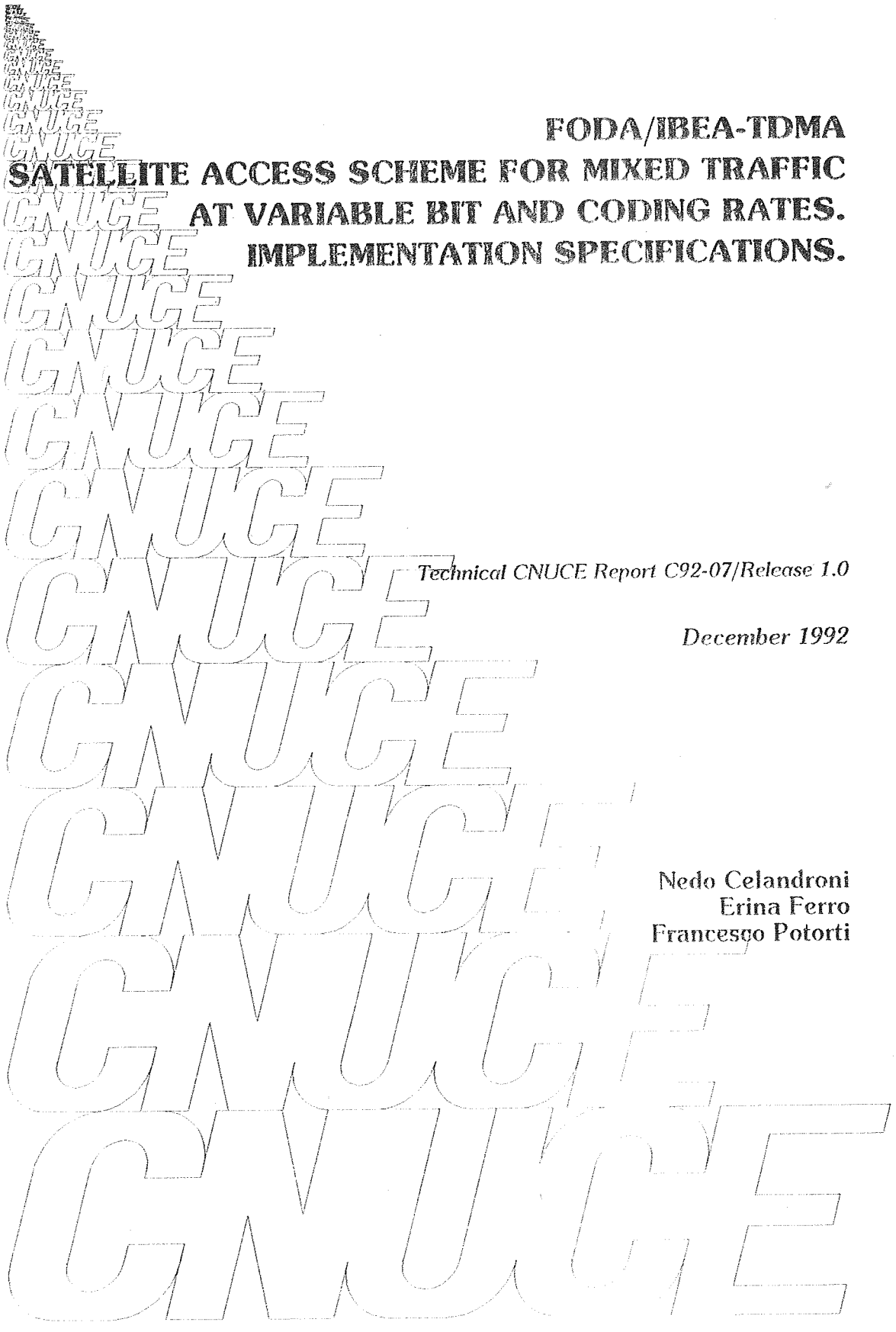
FODA/IBEA-TDMA SATELLITE ACCESS SCHEME FOR MIXED TRAFFIC AT VARIABLE BIT AND CODING RATES. IMPLEMENTATION SPECIFICATIONS.

Technical CNUCE Report C92-07/Release 1.0

December 1992

Nedo Celandroni
Erina Ferro
Francesco Potorti



FODA/IBEA-TDMA
Satellite access scheme for mixed traffic at variable bit and coding rates.

Implementation specifications

Nedo Celandroni (#)
Erina Ferro (#)
Francesco Potorti' (+)

(#) CNUCE/CNR Institute
Via S.Maria 36 - 56126 Pisa (Italy)
(+) Telespazio scholarship holder at CNUCE

Tel. : +39-50-593207/593312/593203
Telex: 500371 CNUCE
Fax : +39-(0)50-904051/904052/589354

CNUCE REPORT C92-07 / Release 1.0

December 1992

Work carried out in the framework of the Italian experimentation program with the Olympus satellite, co-ordinated and funded by ASST (Azienda di Stato Servizi Telefonici)/ISPT (Istituto Superiore Poste Telecomunicazioni).

ACKNOWLEDGEMENTS

The authors express their appreciation and gratitude to Telespazio S.p.A., specifically to Mr. Antonio Marzoli and Mr. Maurizio Neri.

TABLE OF CONTENTS

GLOSSARY

#	number of...
BER	bit error rate
CB	channel byte(s)
CBTRS	carrier and bit timing recovery sequence
COM	communication (interrupt)
CRC	cyclic redundancy check
CSB	control sub-burst. It describes the data burst
CSBS	control sub-burst speed
CSW	control sub-word
CW	channel words
DB	data burst
DMP	delay measurement packet
DSB	data sub-burst. Part of the data burst
DSF	datagram sub-frame. Percentage of the frame devoted to contain the datagram allocations
DGA	datagram allocation. It is expressed as a multiple of the 32-bit word.
DGR	datagram request. It is expressed as multiple of 16 32-bit words.
EOF	end of frame (interrupt)
ESUB	extra stream upper bound. For SSF in fade conditions.
FHE	FIFO half empty (Down)
FHF	FIFO halh full (Up)
HDR	header
HW	16-bit word
IB	information byte(s)
ISR	interrupt service routine
ITG	internal traffic generator
IW	information word(s)
LAN	local area network
NSUB	normal stream upper bound. It is the upper bound for SSF
RB	reference burst
RBR	reference burst received (interrupt)
SH	satellite header
SOF	start of frame (interrupt)
SSF	stream sub-frame. Percentage of the frame devoted to contain the stream allocations
STA	stream allocation. It is expressed as a multiple of the 32-bit word
STCH	stream channel. 1 STCH = one 32-bit word
STR	stream request. It is expressed as a multiple of the 32-bit word
TOVH	tail overhead for each data sub-burst
TUW	traffic unique word
UR	user request
UW	unique word
W	32-bit word.
WORD	32-bit word.

*	**the * indicates a field used by the hardware**
STATIONS	**is the number of stations currently supported by the system.**

INTRODUCTION

The CNUCE Report C88-10 describes the implementation specifications of the FODA system at fixed bit rate (2Mbit/s), using the coding rates 1/2 and uncoded (FODA-TDMA). The system in such a version has been patented (patent N. 9373A/89) in march 1989.

The present report describes the implementation specifications of the FODA system redesigned to work at variable bit (8, 4, 2, 1 Mbit/s) and variable coding rates (1, 2/3, 4/5, 1/2). Both the bit and the coding rates may change on a sub-bursts basis. Such a satellite scheme has been called *FODA/IBEA-TDMA* (Fifo Ordered Demand Assignment/Information Bit Energy Adaptive - Time Division Multiple Access). As in the fixed rate version, the FODA/IBEA software is split in two parts: one for receiving data from satellite (DOWN process) and the other one for transmitting data to the satellite (UP process). The DOWN and the UP processes run each on a separated box. The two boxes are called RX-TDMA terminal and TX-TDMA terminal respectively. Together they constitute the *TDMA terminal.*

The TDMA terminal, realized by the GEC-MARCONI Research Centre in U.K., is a new hardware prototype [10, 11, 12, 13], different from the one used in the fixed rate version of the system. The TDMA terminal features a burst-mode modem and a TDMA controller. The burst-mode modem is digitally implemented and provides a variable bit rate from 1 up to 8 Mbit/s.
The transmit unit of the TDMA controller consists of 3 boards: the user interface card, the control processor (Motorola 68030 running at 25 MHz) and the modem interface card, including the FEC encoder.
The receive unit consists of 3 corresponding boards plus the FEC decoder. The two (receive and transmit) units are interconnected via a SCSI link.

The CNUCE Report C91-21 [7] is required for a correct understanding of the communication between the system and the outside environment. This report contains the description of the communication protocol (GA-FO protocol) between the FODA/IBEA system and the rest of the world.

In the FODA/IBEA access scheme the time is supposed to be divided into transmission windows (slots) in which a number of simultaneously active stations alternate the use of the entire capacity of the channel. The assignment of the time slots is made dynamically upon demand of the users (i.e. the earth stations). Requests for datagram and for stream slots are managed by the system in different ways as different are the characteristics of the two supported types of traffic (isochronous and anisochronous). One of the active stations plays the role of channel dispatcher (master). Techniques to replace the master station in case of fault are provided in order to cause the minimum trouble to the other users, in particular to avoid stream traffic interruptions.

The frame structure

The transmission time is divided into frames 20 ms long. The master station sends a reference burst at the beginning of each frame for synchronisation purposes and for distributing control information to all the stations.
Each frame contains:

-One reference burst (RB), at the beginning of each 20 ms frame, sent by the master station. It contains the time allocations for the transmissions of the requesting stations. The RB is always sent at 2Mbit/s, 2/3 coded.

-One or two transmissions slots per station, containing stream and/or datagram data. Only one transmission slot is assigned to each requesting station for sending stream and datagram data together, if the transmission characteristics of the two types of data are compatible. Otherwise, two consecutive transmission slots are allocated, one for stream data and the other one for the datagram data. Once a station has obtained a stream assignment, it maintains the assignment until it is released by the user. The datagram assignment is not guaranteed to be granted in each frame.
In each frame 2 small control slots (for sending updates of the requests) are assigned on a round-robin scheme basis to the first two active stations which have no assignment in that frame. The control slots are assigned at 2Mbit/s. If all the requesting stations have assignments, the control slots are assigned in any case to the first two stations in scheduling.

-Once every 32 frames, one FAS (First Access Slot) slot is allocated. It is devoted to stations which want to enter the system for the first time. New stations access it in contention mode at 2Mbit/s. It is as large as the two control slots which are not present when the FAS slot is present. The FAS slot is in fixed position inside the frame (before the end of the frame).

The data burst structure

Each station transmits a TDMA data burst in the slot(s) it has been assigned inside the frame. The TDMA burst structure has the format shown in figure 1.

Fig. 1. TDMA FODA/IBEA burst structure

PREAMBLE	Preamble sub-burst. It contains the Carrier and Bit Timing Recovery Sequence (CBTRS).
TUW	48 bits. Traffic Unique Word.
CSB	Control Sub-Burst. It is always 2/3 FEC encoded (to provide maximum protection). The CSB enables the identification and the decoding of all the following sub-bursts.
DSB	Data Sub-burst. It can have any length, up to 4096 words. The length is specified in the CSB. Each DSB has an associated control sub-word (CSW) and a satellite header (SH), included in the CSB (see section 1).

Miscellaneous concepts

Burst overhead indicates the sum of the inter-bursts guard time + the preamble + the unique word + the control sub-burst + the data sub-bursts inter space + the TOVH field of each sub-burst (see section 1).

Stream sub-frame (SSF) indicates the percentage of the frame devoted to contain the stream allocations. In the same way, **datagram sub-frame (DSF)** indicates the percentage of the frame devoted to contain all the datagram allocations.

The request for a stream allocation is made only once by a station and, if accepted by the master station, it is considered valid until the slave station sends a relinquish indication (stream request = 0) or it is declared dead. The **stream request** is made as a multiple of the 32-bit channel word (corresponding to a stream throughput of 1600 bit/s at 8 Mbit/s). The requests for stream allocations are granted by the master in the same order they are received up to an upper boundary in the frame.
Each station sends a **datagram request** (DGR) per frame. DGR is expressed as multiple of 16 32-bit words. The DGR of all the stations are examined by the master station which computes the datagram assignments and broadcast them in the reference burst. Each station computes its datagram request by the formula:

$$DGR = BACKLOG + TRAFFIC * H$$

where BACKLOG is the amount of data lying in the station waiting to be transmitted to satellite, TRAFFIC is the instantaneous traffic coming into the station from the various applications and H is a temporal constant, tuned-up at 0.4 s for the FODA/IBEA system.
The formula is aimed to alleviate the problem of a station having its datagram traffic backlog grown because of the delay occurring between the allocation request made by the station and the allocation assignment made by the master (~500 ms). This problem has been studied in [3]. The algorithm used by the master to compute the datagram allocations is essentially a scaling of the requests by a fixed constant, i.e. each station receives an allocation proportional to its DGR. The allocations are up-bounded (maximum 50% of the DGR), to avoid that a single station takes too much (or the whole) of the datagram sub-frame. They are also down-bounded (minimum 10% of the DGR) in order to allow space for the overheads. Indeed, the datagram requests do not include any overhead.
To better optimise the use of the satellite channel, stream and datagram data (if both present) are transmitted in the same transmission slot (saving preambles and control sub-bursts) if the preamble bit rate of the datagram CBS is lower than the one of the stream CSB.

One of the major problems for communication satellites operating at frequencies above 10 GHz is the high level of rain attenuation encountered. The basic principle, used by the present system, to cope with different levels of the signal attenuation is the variation of the coding and bit rates. The effect on the system performance is, of course, a reduced overall channel capacity in those periods where some fade is experienced at one or more stations.

The stations running FODA/IBEA support the *up-link power control* feature in addition to coding and bit rate control. This feature allows a station to maintain a constant signal level at the satellite input for small up-link fades. At least two differently powered stations can be envisaged: a high powered tube-amplifier station capable of an up-link power control in the range of 0.5 - 50 Watts, and a low-powered solid-state amplifier station working in the 0.5 - 5 Watts. These values are indicative and depend on budget and high-frequency progresses.
The idea of basis in the utilisation of the up-link power control is that no redundancy needs to be applied to the data in order to compensate for up-link fades until the station transmits at full power. In this way, more powerful stations will use (on average) less bandwidth and will be able to counter deeper fades.

In the exchanging of data between two applications in an internetworking system (satellite link included) the most significant parameters characterising the quality of the requested service are:

- the typical data delay
- the maximum jitter of the packet inter-arrival time

- the bit error rate (BER).

In the FODA/IBEA system, the first two parameters are, not negotiable. In case of stream, the delay depends on the round trip time (which, in its turn, depends on the geographic position of the station) plus a fraction of the time frame (roughly 300 ms). The maximum jitter can be assumed one time frame long. In case of datagram both the delay and the jitter depend strongly on the overall traffic conditions of the system and by the saturation control mechanism. In no case, anyway, these parameters can be specified by the application.
The BER, on the contrary, must be specified by the application to choose the quality of the transfer. In the present system, a requested range of BER is specified by means of a class of service (COS) value, as sown in Table 1.

COS	BER not >	BER not <	Example of type of data
1	10^{-8}	----	reference burst; broadcasted control inf.; control sub-bursts; headers
2	$3 * 10^{-7}$	10^{-8}	reliable data: bulk data; special voice/video; interactive data
3	$3 * 10^{-5}$	$3 * 10^{-7}$	standard voice/video
4	10^{-3}	$3 * 10^{-6}$	degraded voice

TABLE 1. The classes of service

Each stream application must send a request indicating the *number of requested channels Cr*, the *minimum acceptable number of channels Cm* (which may coincide with Cr), and the *requested class of service COS*.. Each station of the FODA/IBEA system assembles together all the Crs coming from different stream applications, sending to the master a global unique request, sum of all the Crs. The master grants all the global requests, coming from the active stations, up to a fixed upper bound of the frame devoted to the stream assignments (SSF). If the master grants exactly the requested number of channels, the requesting station provides to distribute the assigned channels among the applications according to the individual requests.
The master is not aware of the parameters Cr and COS of the single applications. They are used by the requesting stations when in fade. For datagram applications, the COS parameter must be specified for each data packet.

Let us fix the reference value of Eb/No as the ratio allowing the system to receive uncoded data with the highest class of service (cos=1, corresponding to BER >= 10^{-8}) at the maximum speed rate (8 Mbit/s). The resulting theoretic value is 12 dB. The fade level is defined as the degradation of the Eb/No ratio, expressed in dB, with respect to the reference value.

Let us define the *redundancy information factor Ri* as the ratio between the time necessary to send a certain amount of information at a certain bit and coding rates and the time necessary to send the same amount of information at the maximum speed (8 Mbit/s), uncoded. *Ri is the product of the coding redundancy Rc by the speed redundancy Rs,* the former being the inverse of the coding rate, the latter being the ratio between the maximum speed and the actual speed. When the necessity arises to compensate an increased attenuation of the signal (i.e. keeping the same value of the BER), it is preferable to increase the factor Rc first, taking advantage of the coding gain, in order to limit the increment of the factor Ri. The speed will be reduced only if the signal to noise ratio (Eb/No) will drop below a

minimum value, necessary to allow the acquisition of the burst modem in a reasonable number of symbols.

The compensation of fade conditions is translated into a request for a wider bandwidth both by the station in fade and by all the other stations wishing to send data to the station in fade. The FODA/IBEA system reacts to the increased requests of bandwidth, due to fade conditions, in different ways according to whether the type of traffic is stream or datagram.
If the minimum value of Eb/No is limited to 4 dB, the speed results to be the same for all the classes of service.

Each station down-link fade level is broadcasted by the master in the reference burst. When station A needs to transmit to station B, the needed data redundancy depends on the station A up-link fade level (known to the transmitting station), on the station B down-link fade level (read by station A in the RB) and on the station A up-link power control capabilities. When multicast or broadcast transmission is made, the highest down-link attenuation among all the receiving station is taken into account.
When data redundancy is introduced to counter a fade condition, the need for greater bandwidth arises. The stream traffic is privileged. A preliminary phase is necessary during which a certain bandwidth for the stream allocation is negotiated between the master an the requesting station. Once a stream request has been granted, the transmitting station attempts to keep unchanged the link characteristics, even in presence of fade conditions. The transmitting station makes a new stream request in order to accommodate the enlarging of the stream data and the master gives to the station a larger allocation. The sum of all the stations stream allocations (SSF), however, cannot exceed certain limits. Specifically, if a station asks for more stream bandwidth in order to satisfy new applications (**normal requests**), the request will be satisfied by the master provided that the SSF does not exceed the **NSUB (Normal Stream Upper Bound)** value. If a station asks for additional stream bandwidth in order to maintain the COS of already set up applications under new fade conditions (extra requests), the request will be satisfied by the master provided that the SSF does not exceed the **ESUB (Extra Stream Upper Bound)** value.

If the enlarging of the stream allocation space is not sufficient to satisfy the increased request, the system notifies to the application the necessity of reducing its bandwidth (up to the Cm value). This is only possible for compressible applications whose Cm value is less than Cr. After receiving the ack from the application, the station begins sending data with reduced bit rate, while maintaining the BER specified by the requested class of service. Applications compressible in bandwidth may include voice and video with variable rate codecs. Incompressible applications are not guaranteed to maintain the requested BER under heavy fade conditions, especially if experienced when the system is strongly loaded.

As far as the datagram is concerned, fading conditions cause an increment of the backlog and of the amount of instantaneous traffic. This automatically increases the station request UR at the sending station site, resulting in an attempt to get more bandwidth. Due to the increasing of the request itself and/or to the possible compression of the datagram allocation space to grant more stream channels, the overall capacity of the datagram may be sensibly reduced, particularly under heavy load conditions of the system. An efficient action of the channel saturation control system is requested to avoid congestion.
A rather simple method to avoid saturation is to block the growth of the backlog for a while, exercising a backpressure on the remote users, when a dangerous situation is detected. Since data are collected from high speed networks (typically LANs provided with flow control mechanisms), the only effect of this procedure is to slow down the overall traffic coming from the remote hosts for a convenient interval of time.

Datagram request algorithm

At initialisation:

- set r_curr_dgr_csbs to 8 Mbit/s (in the Global area);
- set r_prev_dgr_csbs to 8 Mbit/s (in the Global area);

at every data packet reception from the LAN:

- set r_curr_dgr_csbs to the CSB speed required by the received packet, if less than the current r_curr_dgr_csbs value;

at any DGRAM_TIMEOUT:

- set r_prev_dgr_csbs to r_curr_dgr_csbs);
- set r_curr_dgr_csbs to 8 Mbit/s;

at any datagram request (one per frame):

- set the datagram request to the minimum between r_curr_dgr_csbs and r_prev_dgr_csbs.

SECTION 1. THE CONTROL SUB-BURST (CSB)

The control sub-burst conveys information about each of the data sub-bursts (DSBs) in the burst and optionally a number of extra words which could be used to convey status information about the transmission conditions of the burst. The minimum length of the CSB is 5 words.

The control sub-burst is always sent to the station(s) relevant to the associated data and to the master station for the requests updating.

The format of the CSB is shown in Fig. 2; the numbers represent the bit length of each field.

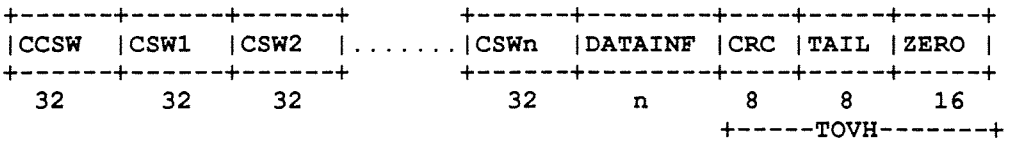


Fig. 2. CSB format

where:

CCSW CSB Control Sub Word. The format is shown in Fig. 2.1.
The two information fields (CSBL and NDSB) of the CCSW field are required by the hardware to recover the CSB.

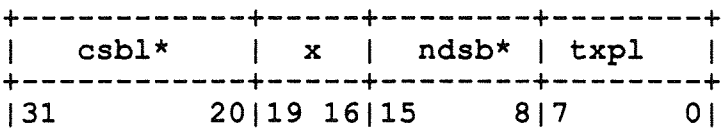


Fig. 2.1 CCSW bit format

csbl 12 bits (bits 31-20). It specifies the length of the CSB in words following the traffic unique word (TUW). This length includes the CCSW, each CSWi and the number of words in DATAINF. It is possible to specify up to 4095 words in the CSB following the TUW. The length does not include the TOVH.

x spare bits (bits 19-16).

ndsb 8 bits (bits 15-8). It specifies the number of DSBs in the burst following the CSB and is required by the receive hardware to decode the CSW for each one, so that the DSBs may be properly received. If *ndsb* is zero, there are no CSWs and this burst therefore consists of the CSB only. Subtracting a word for the CCSW from *csbl*, it is therefore possible to transfer up to 4094 words of information in the DATAINF field, effectively making that field a data transfer channel at the specified parameters of the CSB.

txpl 8 bits (bits 7-0). The power level used to transmit the burst is computed as *(63-txpl)/2* dB.

CSWi 32 bits. Control sub-word for the sub-burst #i. Together with SHi, it conveys all information required for the correct station to receive the information and output it to the appropriate destination. The format of the CSW is shown in Fig. 2.2.

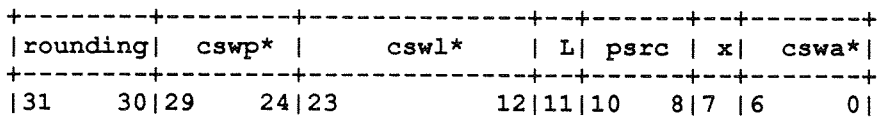


Fig. 2.2 CSWi format

where:

rounding 2 bits. Number of bytes added to round the data to the word.

burst length This number must be subtracted from the data sub-specified in the *cswl* field of the relevant csw.

cswp 6 bits (bits 29-24). Control Sub-Word Parameters. Fig. 2.2.1 shows the format.

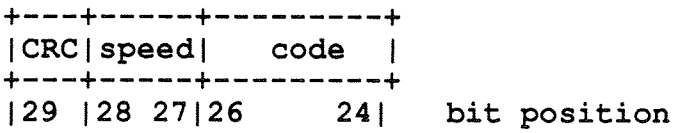


Fig. 2.2.1 cswp format

CRC 1 bit. CRC parity on/off control bit. It determines when a CRC checkword is included in the TOVH.

speed 2 bits. Data sub-burst bit rate.
00 means 1 Mbit/s
01 means 2 Mbit/s
10 means 4 Mbit/s
11 means 8 Mbit/s.

code 3 bits. Data sub-burst coding rate.

000	uncoded
001	uncoded
010	uncoded
011	uncoded
100	1/2 coded
101	2/3 coded
110	4/5 coded.

cswl 12 bits (bits 23-12). Length of the relevant DSB in units of uncoded words.

L — **LCRC** bit. If ON, data with wrong CRC from the LAN.

psrc — 3 bits (bits 10-8). Port source of the data in the relevant DSB. The values are:

000 CSWL_P0* (G703 port A)
001 CSWL_P1* (G703 port B)
010 CSWL_P2* (RS 422)
011 CSWL_ETH* (Ethernet source)
100 CSWL_MTG_ETH (MTG source connected via Ethernet)
101 CSWL_ITG (internally generated data)
110 CSWL_DMP (delay measurement packet from ITG)

x — spare bit.

cswa — 7 bits (bits 6-0). Control sub-word address. It contains the destination address of the relevant sub-burst.

ADDRESSING:

If A6-A4 = 000, broadcast destination. In this case, if A3-A0 = 0, real global broadcast. If not zero, identification of the users group for multicast transmissions. **Multicast is not handled by the hardware.**
Therefore:

0 real broadcast
1-15 user group identifier
16-96 station address.

DATAINF — 0-4094 words. This area is included as part of the CSB to allow transmission of information pertinent to this data burst. Up to 4094 words are allowed in this field. The length of DATAINF is obtained as CSBL-NDSB-1, where the extra word accounts for the CCSW. The format of this area is in Fig. 2.3.

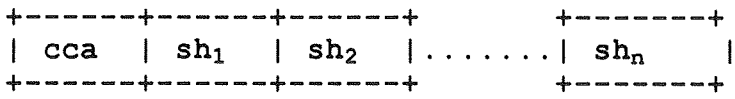


Fig. 2.3 The DATAINF format

where:

cca — Channel Control Area. Always present. See 1.1.

sh_i — satellite header describing the data sub-bursts #i. See 1.2.

CRC 8 bits. This field is only transmitted if the CRC bit in the CSWP field is set ON, otherwise the TAIL is sent, and the ZERO field is 3 bytes long.
Always used in the FODA/IBEA system.

TAIL 8 bits. It is required to flush the encoder.

ZERO 16 bits. It is required to send an integer number of words.

1.1 The Channel Control Area (CCA)

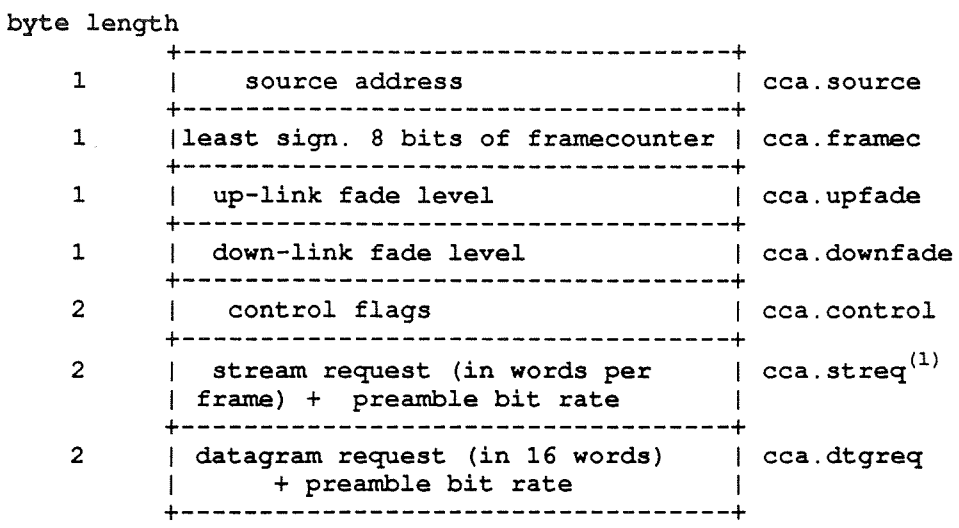


Fig. 3. The CCA format

(1) The stream request must be comprehensive of the burst overhead.

1.1.1 The CCA sub-fields

1.1.1.1 The cca.control field

The format is in Fig. 3.1.

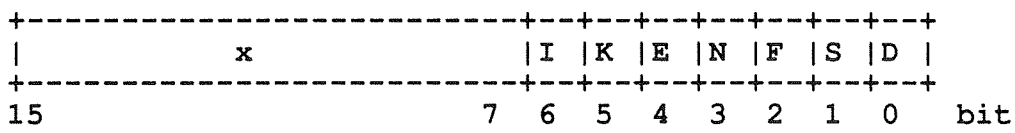


Fig. 3.1 The *cca.control* field format

x spare bits.

I IS_POSSIBLE_MASTER bit. If ON, this slave can become the new master station in case of fault of the current master. If OFF, the station cannot assume in future the role of master. Same bit defined in 6.1.

K IS_SYNCHRONISM bit. The current station offset must be updated in order to maintain the synchronism.

E IS_EXTRA_SREQ bit. This bit can be ON only together with IS_STREQ bit. Additional stream channels are requested not for new applications but only for already existing applications (fade changed, application dead). The master station will use the ESUB boundary

instead of the NSUB boundary in order to try to maintain the already set-up stream sessions.

N IS_NEWBORN bit. New born control message present.

F IS_FADECHANGED. The fade condition for the station has changed.

S IS_STREQ bit. Stream request present.

D IS_DGREQ bit. Datagram request present. A datagram request equal zero is considered as an hello message.

1.1.1.2 The cca.streq and the cca.dtgreq fields

Both of them have the format in Fig. 3.2.

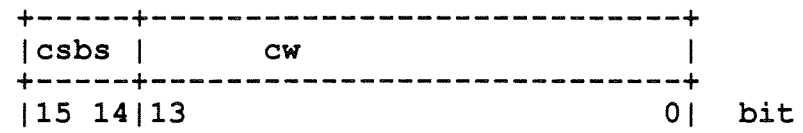


Fig. 3.2 cca.streq and cca.dtgreq format

csbs 2 bits. Control sub-burst speed. Minimum bit rate required for the CSB. It is computed according to the algorithm described in the Introduction for the datagram request.

cw 14 bits. Number of requested channel words .
The ***stream request*** is expressed in words per frame (maximum stream throughput which can be expressed in 14 bits = 25.6 Mbit/s when the transmission bit rate is 8 Mbit/s). It must contain the burst overhead (expressed in number of words).
The ***datagram request*** is expressed as multiple of 16 words per frame. It is the DGR as defined in the *Miscellaneous Concepts* section of the *Introduction*.

1.2 The satellite header SH

TTwo different satellite header formats are defined for stream and for datagram data. The satellite headers are contained in the DATAINF portion of the CSB, following the CCA. Each SH is associated with a DSB in the same burst.

The satellite header for stream data is 40 bits long. The format is in Fig. 4-A.

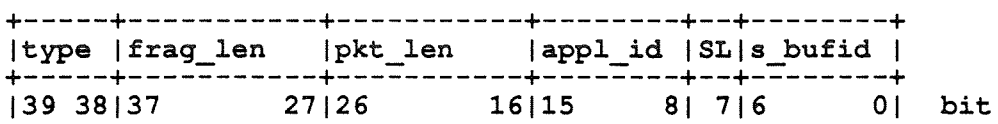


Fig. 4-A. The satellite header format for stream data

where:

type 2 bits. Type of the satellite header.
SH_STREAM = 1 (satellite header for stream data).

frag_len 11 bits. Byte length of the first fragment.

pkt_len 11 bits. Byte length of the packets.
It must be the same for all the packets of the same buffer.

appl_id 8 bits. Identifier of this stream application inside the FODA/IBEA system.

SL S_LAST bit. Last fragment bit. If ON, the first fragment is the last for his buffer.

s_bufid 7 bits. Buffer identifier for stream data.

The satellite header for datagram data is 16 bits long. The format is in Fig. 4-B.

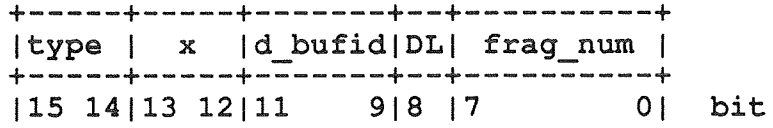


Fig. 4-B. The satellite header format for datagram data

where:

type 2 bits. Type of the satellite header.
SH_BULK = 2 (satellite header for bulk data)

SH_INTER = 3 (satellite header for interactive data)

x 2 spare bits.

d_bufid 3 bits. Buffer identifier for the datagram data.

DL D_LAST bit. Last fragment bit. If ON, this is the last fragment for his buffer.

frag_num 8 bits. Number of this fragment inside this buffer identifier.

SECTION 2. THE STRUCTURE OF THE BUFFERS

2.1 On the UP side

Data enqueued to the UP process from the LAN are organized in buffers. The format of the buffers on the UP side is shown in Fig. 5-A.

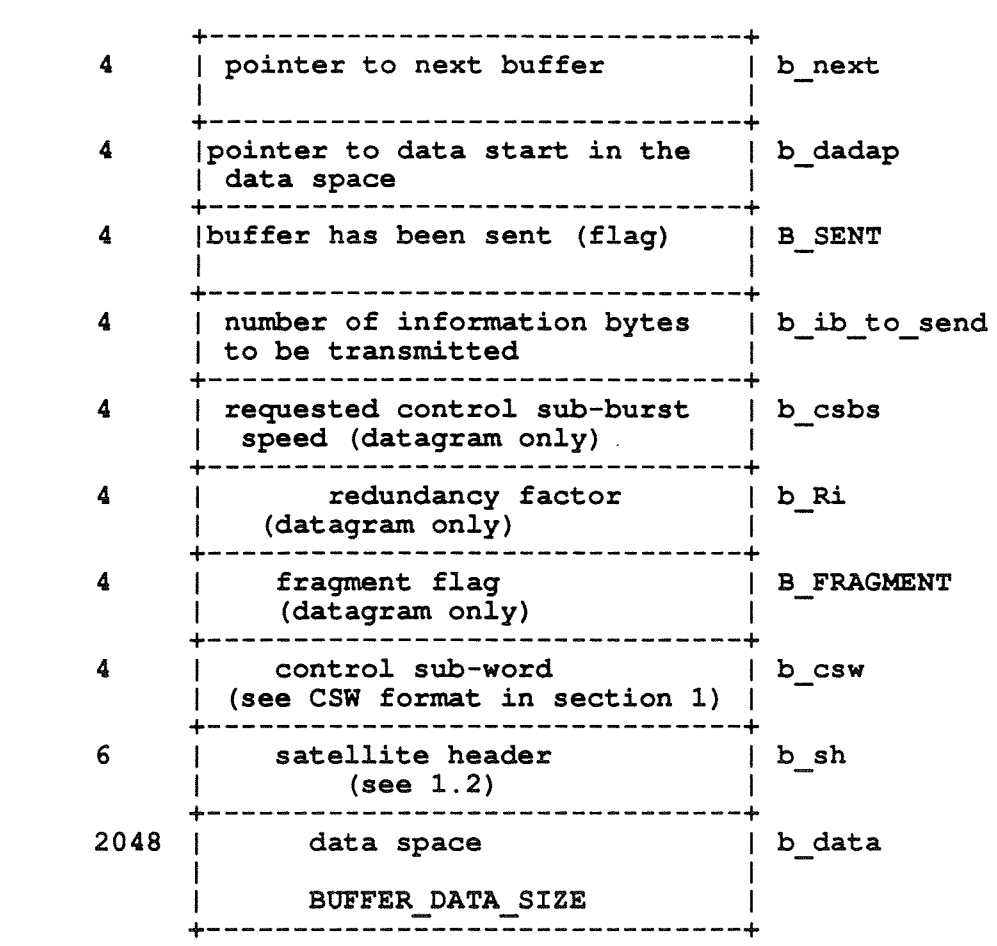


Fig. 5-A Buffer structure on the UP side (*up_buffer*)

Free buffers on the Up process are mantained in the ***free_buffers*** queue.

2.2 On the DOWN side

Data received from satellite are sent to the LAN. They are contained in buffers having the format shown in Fig. 5-B.
PACKET means a complete message. It may consists of more than one fragment , i.e. more than one buffer.

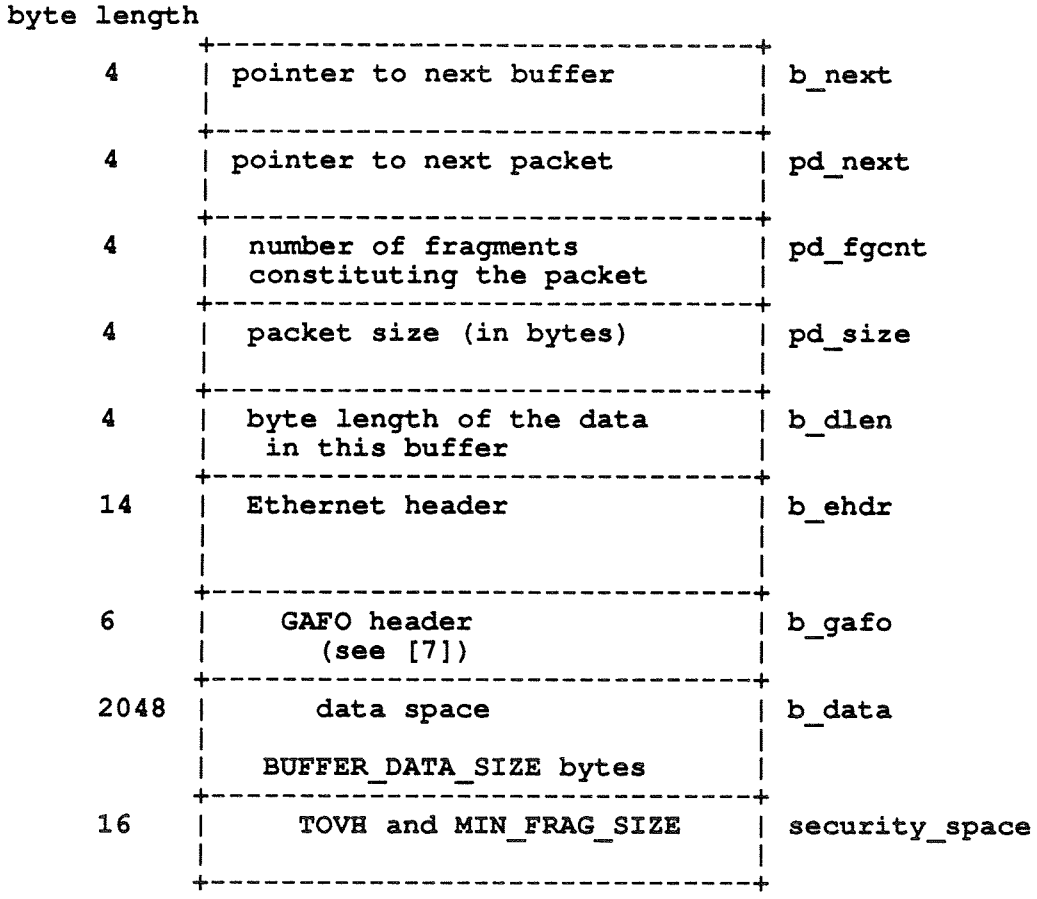


Fig. 5-B Buffer structure on the DOWN side (*dn_buffer*)

Free buffers on the Down process are mantained in the ***free_buffers*** queue.

SECTION 3. AREA USED FOR THE SCSI MESSAGE EXCHANGE

Messages exchange between the Down and the Up sides of the FODA/IBEA software are currently performed by means of a SCSI connection between the controllers. Due to the high cost of the start of the SCSI DMA operation, **only one SCSI message per frame is sent** from the DOWN process toward the UP process, **at the reception of the reference burst.** The number of transferred bytes is irrelevant from the operation cost point of view. Therefore, a single memory area is prepared in each frame by the DOWN process. This area, called **C A** (Communication Area) contains the information shown in Fig. 6.

byte length		
	ref.burst descriptor of the last received reference burst	rb_got
	reference burst to be transmitted if master station (see 3.1)	rb_built
*	fade level of each station	st_dBl
1	delay measurement packet present	dmp_present
1	Up frame number at DMP creation	dmp_frnum
2	Up timer value at DMP creation	dmp_timer
2	Up backlog at DMP creation	dmp_backlog
1	up-fade changed in my station	MY_UP_FADE_CHANGED
1	down-fade changed in my station	MY_DN_FADE_CHANGED
1	my up-link fade value [dB]	my_upfade
1	my down-link fade value [dB]	my_downfade
4	flag to indicate that the station is master	IS_MASTER
16	data from operator interaction	operdata

Fig. 6. The CA area for SCSI communication

* This field is STATIONS byte long.

3.1 The reference burst descriptor (rb_got) of the last received RB

The **rb_got area** has the format (sized in bit) shown in Fig. 6.1.

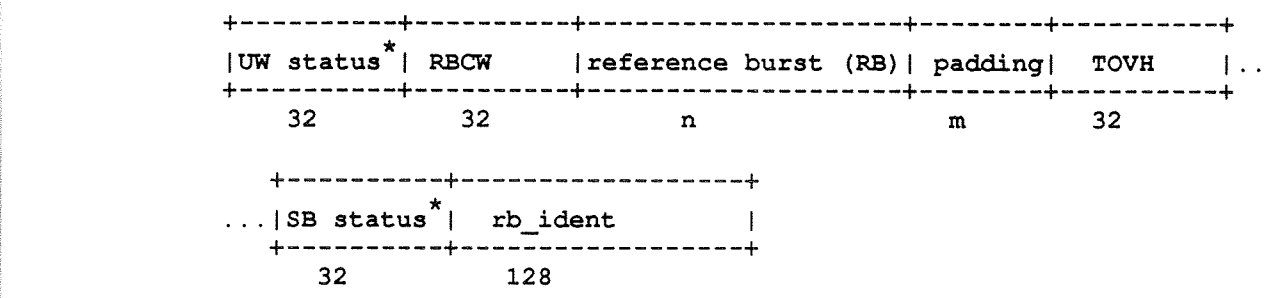


Fig. 6.1. The RB received from satellite (ca.rb_got)

where:

UW status 32 bits. **uwstat** member. Provided by the hardware of the DOWN side.

It has the bit format shown in Fig. 6.1.1.

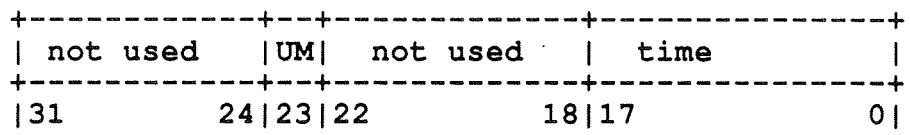


Fig. 6.1.1 The UW status

UM UW miss bit. If ON, all following is invalid.

time UW position within the frame relative to the UW reception (frame clock).

RBCW 32 bits. See 3.2.

RB data of the reference burst. See 3.2.

padding used to make the RB length an integer number of words.

TOVH 32 bits. CRC+TAIL+ZERO.

SB status 32 bits. **sbstat** member. Filled by the hardware of the DOWN side. Common to any sub-burst. The format is shown in Fig. 6.1.2.

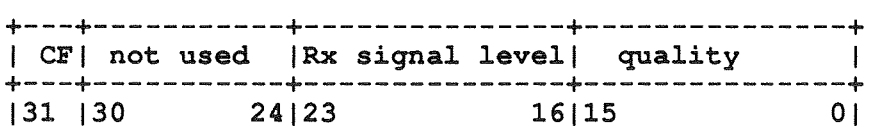


Fig. 6.1.2 The SB status

where:

CF	1 bit. CRC fail (if ON).
RX signal level	8 bits. Power level in receiving this DSB.
quality	16 bits. Down-link fade of the receiving station.

rb_ident 16 bytes. An ASCII string to identify the reference burst. Used for debug purposes.

The RB is always sent 2/3 coded at a bit rate of 2 Mbit/s.

The first word of the reference burst is the **RBCW** field, whose format is shown in Fig. 6.2.

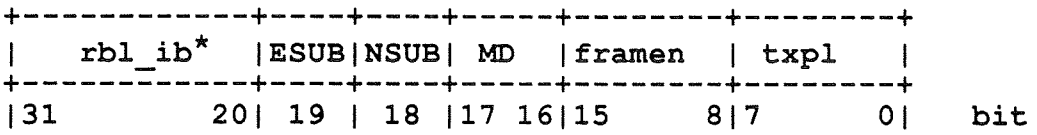


Fig. 6.2. The RBCW format

rbl_ib 12 bits. Reference burst byte length (information bytes).

ESUB 1 bit. Extra stream upper bound. The SSF reaches the ESUB boundary. Neither the stream requests enlarged due to the fade conditions can be accepted. When ON, also NSUB is ON.

NSUB 1 bit. Normal stream upper bound. The SSF reaches the NSUB boundary. No more normal stream channels are available to be assigned.

MD 2 bits. Master is going down. The new master must start to send the RB. The current master will cease after further 14 RBs. The two bits are set to 01 14 frames before the master goes down, then to 10 in the next frame and to 11 from there on. The new master will set them to 00.

framen 8 bits. Number of frame this reference burst refers to. When the last 5 bits are zero (every 32 frames), this frame contains the First Access Slot (at the end of the frame) instead of the 2 control slots.

txpl 8 bits. Transmit power level (in dB) to send the reference burst.

Immediately following the RBCW there is the **reference burst data**, whose format is in Fig. 6.3.

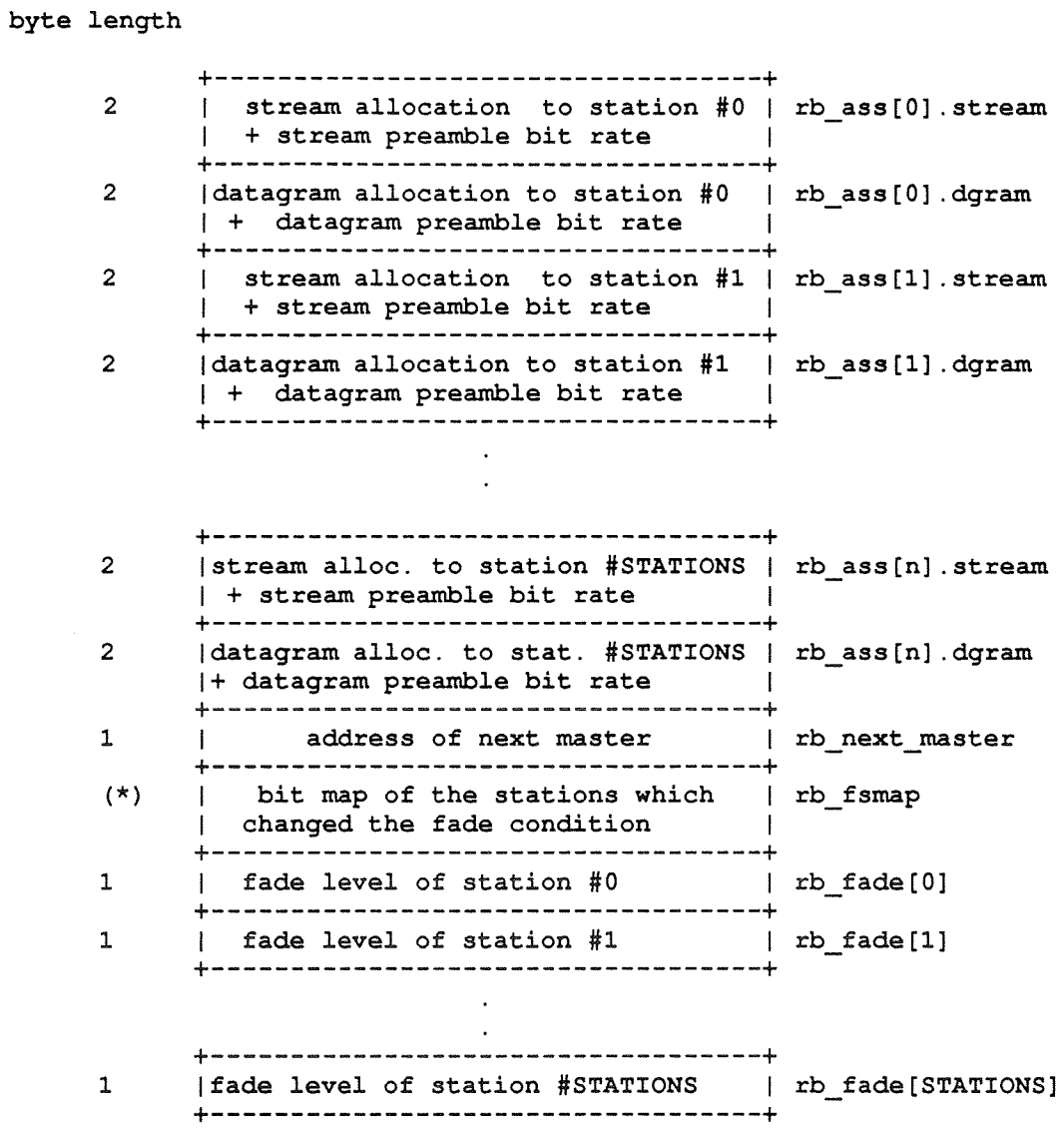


Fig. 6.3 The reference burst data

(*) The size of this field is: 8 * [(STATIONS + 7) >> 3] bits.
One bit for each existing station in the system.

The station address is used as offset (real address decremented by 16) inside the allocation part of the reference burst to get the correct allocation. As the datagram assignment may or may not be present and, in any case, it can be different from the previous assignment, the transmission window of each station must be recalculated for each frame. If the reference burst is not received, no transmission window can be set up.

If a station has no allocation, the relevant *ass.stream* and *ass.dgram* members contain zero.

One byte is devoted to contain the fade level of each station (fade). fade[n]= 0xFF means that the station having address [n] is not active.

Each of the ***ass.stream*** and ***ass.dgram*** members has the format in Fig. 6.3.1

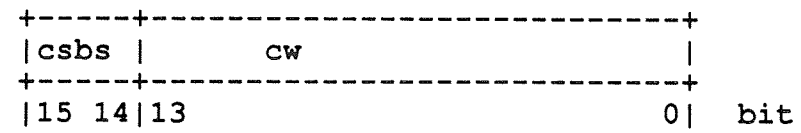


Fig. 6.3.1 The rb.ass[n].stream and rb.ass[n].dgram formats

csbs 2 bits. Control sub-burst speed.
Bit rate used for transmitting the CSB of the relevant data.

cw 14 bits. Channel words. Number of words assigned on the channel.

SECTION 4. AREAS PRESENT IN THE *UP* PROCESS ONLY

4.1 The statistics area

Statistics on the UP process side are collected in a statistic area, whose format is shown in Fig. 7. All the fields are counters.

byte length		
4	counter of the Start of Frame (SOF) interrupts	sof_count
4	counter of the Communications (COM) interrupts	com_count
4	counter of the Fifo Half Full (FHF) interrupts	fhf_count
4	times the SOF cannot found a transmit program	sof_synch_losses
4	times the SOF did not executed the transmit program	Drea_synch_losses
4	number of packets received with unknown code	wrong_packets
4	number of fatal errors in receiving from the LAN	lan_fatal_err
4	missed packets in receiving from the LAN	lan_missed_err
4	framing errors in receiving from the LAN	lan_fram_err
4	CRC errors in receiving from the LAN	lan_crc_err
4	number of good stream packets received from the LAN	lan_strm_good
4	number of dropped stream packets in reception from the LAN	lan_strm_dropped
4	number of good interact. packets received from the LAN	lan_intr_good
4	number of dropped inter. packets in reception from the LAN	lan_intr_dropped
4	number of good bulk packets received from the LAN	lan_bulk_good

4	number of dropped bulk packets in reception from the LAN	lan_bulk_dropped
4	total packets received from the LAN	lan_rx_total
4	number of packets whose Tx was deferred	lan_deferred
4	number of packets sent at first try	lan_one_try
4	number of packets not sent at first try	lan_more_try
4	number of total packets transmitted to satellite	lan_tx_total

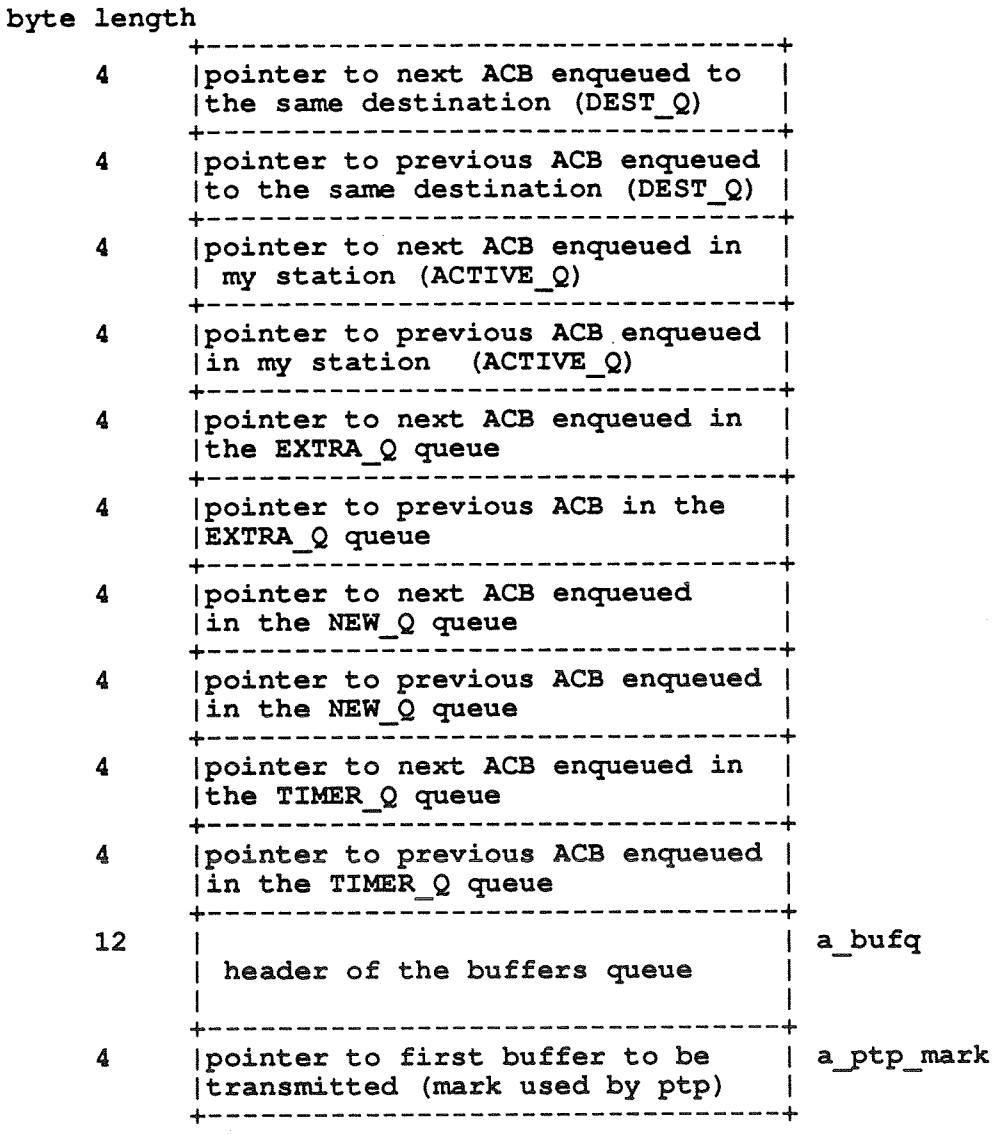
Fig. 7. The format of the TXSTAT area

FODA/IBEA-TDMA. Implementation specs.

4.2 The Application Control Block (ACB)

The format of the request for stream channels, together with the fields meaning, is described in [7]. After receiving a request for stream channels from an application, the UP process builds the Application Control Block (ACB) containing the information described in Fig.8.
The address of the correct ACB is very easy to retrieve. An ACB may be accessed by using either the FODA_appl_id (application identifier assigned by the FODA/IBEA system to a requesting application) or the data destination address.

Up to 64 different applications per station are possible.



4	ACB used by an application	A_USED
4	ACB enqueued in the NEW_Q queue	A_WAIT_IN NEWQ
4	ACB enqueued in the EXTRA_Q	A_WAIT_IN_EXTRAQ
4	ACB enqueued in the TIMER_Q	A_WAIT_IN_TIMERQ
4	compressed application	A_COMPRESSED
4	application waiting for user ack	A_WAIT_FOR_USER_ACK
4	application working out of specs	A_OUT_OF_SPECS
4	application is undead	A_ZOMBIE
1	data destination address	a_dest
1	max # of packets in queue	a_qdepth
1	requested class of service	a_cos
1	application id in the gateway	a_gw_aid
1	application id in the FODA/IBEA	a_foda_aid
1	current buffer identifier	a_bufid
2	requested throughput (w x frame)	a_req_iw
2	minimum acceptable throughput	a_min_iw
2	timeout counter	a_tm_cnt
6	current transmission parameters	a_current
6	next transmission parameters	a_next

Fig. 8. The ACB format

For each station, the ACBs in the EXTRA_Q must be linked together in such a way to leave the applications working out of the specifications at the beginning of the queue. They are followed by the compressible applications and, finally, by the applications working in normal way.

FODA/IBEA-TDMA. Implementation specs.

4.2.1 Some members of the ACB structure

A_ZOMBIE the application is dead but some buffers must be still sent.

a_dest destination address of the stream data generated by the application. See the addressing rules specified in section 1. Real addresses range from 16 to 96, but 16 is subtracted to use this address as an offset.

a_cos requested class of service for the data. This value is used as an offset to get the address of the class of service table, COSTAB, associated to the requested class of service.

a_gw_aid identifier assigned by the gateway to this application.

a_foda_aid identifier assigned by the FODA/IBEA system to this application.

a_req_iw number of stream information words (per frame) requested for this application.

a_min_iw minimum acceptable number of stream information words (per frame). It must be less or equal to a_req_iw. If equal, the application cannot be compressed.

a_tm_out if this ACB is in the TIMER_Q, this is the timer upon which expiration the application is rejected because it did not acknowledge a requested compression.

a_current/_next current/next transmission characteristics.
The format of these two areas is shown in Fig. 8.1.

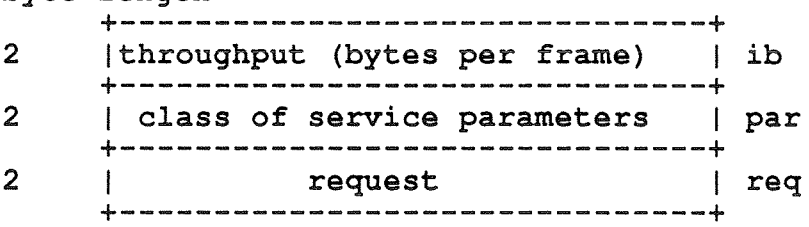


Fig. 8.1. The a_current & a_next format

The field *par* has the following structure:

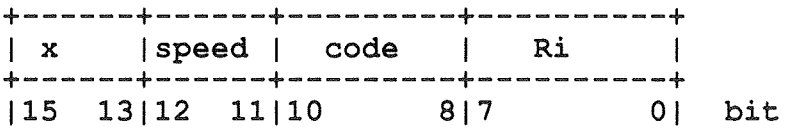


Fig. 8.1.1 The *par* format

where:

x	3 bits. Spare bits.
speed	2 bits. Data speed rate.
code	3 bits. Data coding rate.
Ri	8 bits. Redundancy information. Overall data redundancy factor.

The field ***req*** has the following structure:

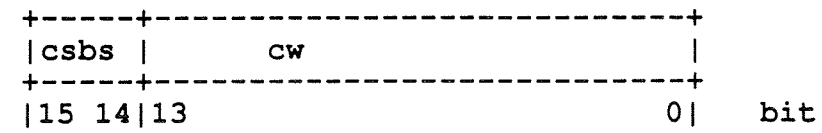


Fig. 8.1.2 The *req* format

where:

csbs	2 bits. Control sub-burst speed.
cw	14 bits. Requested number of channel words.

Each ACB is linked in the D_TABLE (Destination TABLE), to the destination address where the application wants to send data.
D_TABLE has STATIONS+16 entries, one for each possible address. It is accessed by using the station address as offset. D_TABLE is described in the GLOBAL area in 4.3.

4.3 The GLOBAL area

Information of common usage, accessed by different tasks, are maintained in a common GLOBAL area. The area format is shown in Figs. 9 and the member lengths are expressed in bytes.

1) FODA/IBEA active flag

4	FODA/IBEA active flag	FODA_UP

Fig. 9.1

2) My transmission situation

4	frame number	frame_number
4	my TX side is in fade (flag)	FADING
4	station declared dead by the master (flag)	DOWN
4	this station is master (flag)	MASTER
4	this station can become master (flag)	POSSIBLE_MASTER
1	current fade level on my Tx side	my_current_upfade
1	previous fade level on my Tx side	my_prev_upfade
1	address of my station	my_address
6	gateway Ethernet address	my_GW_addr

Fig. 9.2

3) Stream request area

The marked areas (+) have the format as in Fig. 8.1.2.

2	current channel words requested	r_current(+)
2	granted stream channel words	r_granted(+)
2	wanted number of channel words	r_updated(+)
2	current # of overhead cw due to csb	r_curr_csbovh_cw
2	next # of overhead cw due to csb	r_next_csbovh_cw
2	current # of overhead cw due to dsb gaps	r_curr_dsbovh_cw
2	next # of overhead cw due to dsb gaps	r_next_dsbovh_cw

Fig. 9.3

4) Datagram csbs variables

4	datagram stats refresh interval	dgram_refresh
4	datagram csbs in current time period	r_curr_dgr_csbs
4	datagram csbs in next time period	r_next_dgr_csbs

Fig. 9.4

5) Flags

4	SSF boundary NSUB exceeded (flag)	R_NSUB_EXCEEDED
4	SSF boundary ESUB exceeded (flag)	R_ESUB_EXCEEDED
4	stream req. ready to master (flag)	R_STREAM_REQUEST_READY
4	extra request ready to master (flag)	R_EXTRA_STREQ
4	number of stations for user group request	r_group_req_size

Fig. 9.5

6) Traffic queues

See the queue header format in 4.4.

12	bulk messages queue header	bulk_q
12	interactive messages queue header	intr_q
12	active applications queue header	active_q
12	new appl. waiting for grant queue header	new_q
12	old appl. waiting for grant queue header	extra_q
12	waiting for user ack queue header	timer_q
4	pointer to the free buffer area	free_buffers

Fig. 9.6

FODA/IBEA-TDMA. Implementation specs.

7) *Message queues*

4	UPlh main message queue identifier	UPlh_qid
4	LSnd task message queue identifier	LSnd_qid
4	Prnt task message queue identifier	Prnt_qid

Fig. 9.7

8) *Tasks*

4	UP LAN handler task (UPlh) id	UPlhid
4	clear queues task identifier	Clrqid
4	send-to-LAN task identifier	Lsndid
4	down-read task identifier	Dreaid
4	down side simulator task identifier	Dsimid
4	print task identifier	Prntid

Fig. 9.8

9) ***CSB building block***

There are 2 of these structures, called ***first_db*** and ***second_db***, respectively.

10	CCA area (see 1.1)	cca
	data burst control sub-words array (4 * MAX_CSW)	cbb_csw[i][j]
	data burst satellite headers array (4 * MAX_CSW)	cbb_sh [i][j]
4	indices to previous arrays	cbb_index
4 * 4096	contiguous memory buffer for the SHDRs	cbb_buf
4	control sub-burts speed	cbb_csbs
4	channel word length of this data burt	cbb_cw
4	information bytes length of the ccsw	cbb_cswl_ib
	transmit program area (see 4.5) (8 * TP_SIZE)	cbb_tp

Fig. 9.9

10) *Destination Table (d_table)*
Each row from 0 up to 15 (user groups table) has the following format:

12	user group ACB queue header (see queue header format in 4.4)	u_queue
1	user group existing flag	U_UP
1	user group in fade (flag)	U_FADING
1	user group highest fade level	u_hfl
1	number of members in the user group	u_num
8	destination stations addresses (1 byte for each destination)	u_members[]

Fig. 9.10

Each row from 16 up to MAX_ST has the format shown in the following. The row offset corresponds to the station address (by adding 15 to the row offset).

12	stations ACB queue header (see queue header format in 4.4)	s_queue
1	station up flag	S_UP
1	station receiving side in fade (flag)	S_FADING
1	current receiving side fade level	s_current_dnfade
2	bit mask of the user group this station belongs to	s_ugmask
1	previous receive side fade level	s_prev_dnfade
1	master station indication	S_MASTER

Fig. 9.11

11) ***Application Control Block Arrey.***
Area devoted to the ACBs (ACB size * MAX_APPLICATION).
Each ACB has the format shown in Fig. 8.

12) *ACB counter*

4	counter of the ACBs in use	appl_count

Fig. 9.12

13) *Class Of Service Tables*

13) Class Of Service Tables

costab is an array [H_COSTAB_LIMIT] [MAX_FADE].
Each element of such an array is an half-word having the following format:

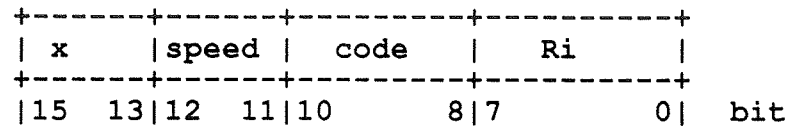


Fig. 9.13

x 3 bits. Spare bits.

speed 2 bits. Data speed rate.

code 3 bits. Data coding rate.

Ri 8 bits. Redundancy information. Overall data redundancy factor.

14) DSB interspace tables

dsb_interspace_carrier_ib
dsb_interspace_carrier_ib
dsb_interspace_cb

are 3 arrays [4] [4]

15) Communication Area (ca). Described in section 3.

4.4 The queues on the UP process

4.4.1 The ACB queues

a) The application control blocks describing stream applications which send data to the same destination are enqueued in a **DEST_Q** queue. There is a DEST_Q for each active station, one for each user group and one for the broadcast.

b) All the ACBs present in a station are enqueued in the **ACTIVE_Q** queue.

c) The ACBs of already working stream applications requesting additional stream channels to counter fade conditions, are enqueued in the **EXTRA_Q** queue to wait for the master granting. Only requests for a larger number of stream channels are enqueued in this queue. Requests for a lower number of stream channels are surely granted by the master. Therefore, such requests are not included in the list of the requests which may or may not be granted!!
The EXTRA_Q is created before to send an extra stream request to the master. It is deleted when the relevant allocation is received.
The EXTRA_Q is also created when a stream application dies and its stream channel words are re-distributed among all the applications which require a change of the allocation. In this case, the global stream request is not sent to the master, but the redistribution is handled inside the station.

d) The ACBs of new applications requesting stream channels for the first time are enqueued in the **NEW_Q** queue to wait for the master granting.

e) ACBs relative to applications waiting for a confirm about the request of changing the number of stream channels are enqueued in the **TIMER_Q** queue.

4.4.2 The buffer queues

a) Free buffers are enqueued in the **free_buffers** queue.

b) Buffers coming from the LAN and containing bulk data to be sent to satellite are enqueued in the **bulk_q** queue.

c) Buffers coming from the LAN and containing interactive data to be sent to the satellite are enqueued in the **intr_q** queue.

d) For each active ACB, the enqueued buffers (containing stream data) form the **a_bufq** queue.

4.4.3 The queue header format

All the queues in the system have the same **queue header format** shown in Fig. 10:

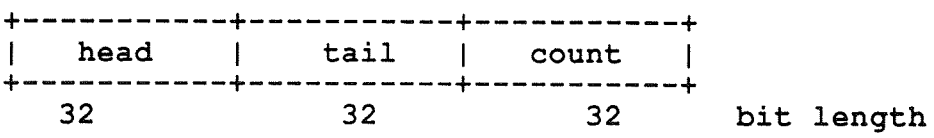


Fig. 10 Header of any queue

head pointer to first element in the queue.

tail pointer to last element in the queue.

count number of elements in the queue.

4.5 The transmit program structure

The transmit program is described in the tpcmd table.
Each row of the table is 64 bits long, having the format in Fig. 11.

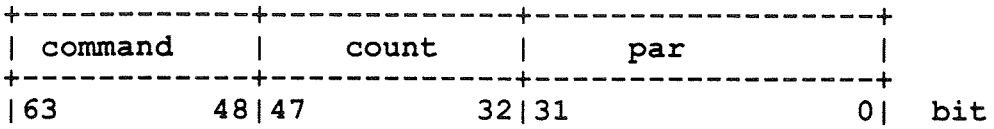


Fig. 11 The transmit program row structure

where:

command command to be issued. It may be:

write_flag_byte;
push_word;
push_null_byte;
push_bytes;
send_event;
set_word;
continue_tp;
end_of_tp.

count counter of bytes or words, according to the command.

par parameter. It may be an address or it may contain data, according to the command.

4.6 The ACK_TIMER_EXP structure

This structure is used in every area used to communicate to the correct ACB the occurred timer event.

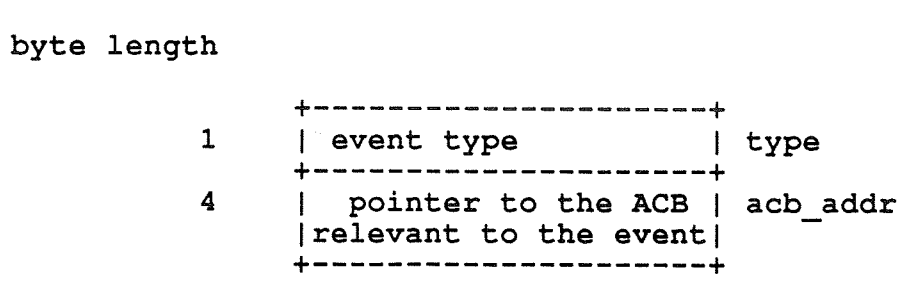


Fig. 12 The ACK_TIMER_EXP structure

4.7 The COSTABs for the 4 classes of service

(a)-Class of service = 0 (BER < 10^{-8})					
C/No range [dBHz]	Eb/No range [dB]	Fade range [dB]	Bit rate [Mbit/s]	Code rate	Data redundancy (+)
81	12	0	8	1	1
80.5 - 77	11.5 - 8	0.5 - 4	8	4/5	1.25
76.5 - 75	7.5 - 6	4.5 - 6	8	2/3	1.5
74.5 - 74	8.5 - 8	6.5 - 7	4	4/5	2.5
73.5 - 72	7.5 - 6	7.5 - 9	4	2/3	3
71.5 - 71	8.5 - 8	9.5 - 10	2	4/5	5
70.5 - 69	7.5 - 6	10.5 - 12	2	2/3	6
68.5 - 68	8.5 - 8	12.5 - 13	1	4/5	10
67.5 - 66	7.5 - 6	13.5 - 15	1	2/3	12

(b) - Class of service = 1 (10^{-8} < BER <3 x 10^{-7})					
C/No range [dBHz]	Eb/No range [dB]	Fade range [dB]	Bit rate [Mbit/s]	Code rate	Data redundancy (+)
81 - 80.5	12 - 11.5	0 - 0.5	8	1	1
80. - 76.5	11 - 7.5	1 - 4.5	8	4/5	1.25
76. - 75	7 - 6	5 - 6	8	2/3	1.5
74.5 - 73.5	8.5 - 7.5	6.5- 7.5	4	4/5	2.5
73 - 72	7 - 6	8 - 9	4	2/3	3
71.5 - 70.5	8.5 - 7.5	9.5- 10.5	2	4/5	5
70 - 69	7 - 6	11 - 12	2	2/3	6
68.5 - 67.5	8.5 - 7.5	12.5- 13.5	1	4/5	10
67 - 66	7 - 6	14 - 15	1	2/3	12

(c)- Class of service = 2 ($3x10^{-7}$ < BER <$3x 10^{-5}$)					
C/No range [dBHz]	Eb/No range [dB]	Fade range [dB]	Bit rate [Mbit/s]	Code rate	Data redundancy (+)
81 - 78	12 - 9	0 - 3	8	1	1
77.5 - 75	8.5 - 6	3.5 - 6	8	4/5	1.25
74.5 - 72	8.5 - 6	6.5 - 9	4	4/5	2.5
71.5 - 69	8.5 - 6	9.5 - 12	2	4/5	5
68.5 - 66	8.5 - 6	12.5 - 15	1	4/5	10

(d) - Class of service = 3 ($3x10^{-6}$ < BER <10^{-3})					
C/No range [dBHz]	Eb/No range [dB]	Fade range [dB]	Bit rate [Mbit/s]	Code rate	Data redundancy (+)
81 - 76	12 - 7	0 - 5	8	1	1
75.5 - 75	6.5 - 6	5.5 - 6	8	4/5	1.25
74.5 - 73	8.5 - 7	6.5 - 8	4	1	2
72.5 - 72	6.5 - 6	8.5 - 9	4	4/5	2.5
71.5 - 70	8.5 - 7	9.5 - 11	2	1	4
69.5 - 69	6.5 - 6	11.5 - 12	2	4/5	5
68.5 - 67	8.5 - 7	12.5 - 14	1	1	8
66.5 - 66	6.5 - 6	14.5 - 15	1	4/5	10

Tab. 2 - Transmission characteristics for various classes of services

(+) Data redundancy = (8/bit rate) x (1/code rate)

FODA/IBEA-TDMA. Implementation specs.

4.8 Communications from/to the external gateway to/from the UP process

The CNUCE Report C91-21 is devoted to explain the communication protocol between the FODA/IBEA system and the GA-FO gateway [7].

GTW_MSG is the area used by the UP process to receive **messages from the GA-FO gateway and to send messages to the GA-FO gateway** from the FODA/IBEA system. The area has the format as described in [7].

GTW_DATA is the area used by the Up process to receive data from the GA-FO gateway for transmission to the satellite and to send data from the DOWN process to the GA-FO gateway on reception of data from the satellite. The GTW_DATA area must be present both in the DOWN and in the UP processes. Again, the format is explained in [7].

SECTION 5. SOME BEHAVIOURS IN THE UP PROCESS

5.1 Distribution of the granted stream allocation among the requesting applications

When a fade condition is experienced, the station needs more stream channel words in order to counter the fade. Specifically, when the station fade condition worsens, the necessary stream channel words are recomputed, the suffering applications (those working with Redundancy information >1 or those compressed) are enqueued in the EXTRA_Q and an extra request is sent to the master.
After a round trip delay, if the granted number of stream channels is equal to the requested one, they are distributed among the requesting applications on the basis of the single requests.
If less, it must be checked if the master has done all its best to satisfy the request or if the request has been lost (because incorrectly received). If the master has not done all its best, the request is re-issued and nothing new is distributed. If all the best has been done, the granted channels must be distributed among (part of) the requestors.

If the necessity of enlarging of a stream application, due to fade conditions, cannot be satisfied and the application is compressible, it is requested to compress itself. A request of compression is only used as a last resort, when it is no possible to further increase the redundancy on the satellite channel while maintaining the uncompressed throughput.

5.2 The history of a buffer on the Up process.

All unused buffers are linked in the **free_buffers queue** There is only one *"link"* pointer as a buffer can reside in only one queue at a time.

The first *RBUFNO* buffers in the free_buffers queue are pointed to by the pointers in the LANCE receive ring. Therefore they are eligible to be filled by the LANCE DMA logic. RBUFNO can be set at the compile time to assume any value power of 2 in the range from 1 to 128. It can be viewed as both the minimum number of buffers that must reside in the free_buffers queue at any one time, and the maximum number of consecutive LANCE interrupts that can be lost without affecting the FODA/IBEA behaviour.

After the first buffer in the free_buffers queue is filled by the LANCE DMA logic with the contents of an Ethernet packet (the data area in the buffer has room enough for any Ethernet packet), the LANCE interrupt service routine—in the following simply **LAN isr**— dequeues the buffer from the free_buffers queue and looks at the first byte of the received packet in order to distinguish between GAFO control messages and GAFO data messages.

If the packet appears to be spurious (not a GAFO packet) it is dropped by simply linking the buffer to the end the buffer_queue, an operation that in the following will be simply referred to as *releasing the buffer.*

If the buffer contains a GAFO control message, the *LAN isr* copies it to a message buffer that is sent to the *UPlh* task queue, then releases the buffer.

If the buffer contains a GAFO data message, the *b_datap* member in the buffer is set to point to the first byte of the data. Then, two cases are distinguished: stream data or datagram data.

Stream data : LAN isr actions	**Datagram data : LAN isr actions**
The packet is dropped if the relevant ACB is either not used, in zombie state, waiting in the NEW_Q, or if enqueueing this packet to the ACB would exceed the maximum buffer queue depth for this application.	The packet is dropped if the destination is down or is a non-existent user group.
Some bit fields in the *b_csw* member are initialised, notably: *CRC* is set to 1; *psrc* is set; *cswa* is set according to the destination; *B_SENT* flag is set off.	Some bit fields in the *b_csw* member are initialised, notably: *CRC* is set to 1; *rate* is set according to the requested *cos* and *destination*; *cswl* is set according to the packet length; *psrc* is set; *cswa* is set according to the destination; *B_SENT* flag is set off.

Some bit fields in the *b_sh* member are initialised, notably: *type* is set to SH_STREAM; *frag_len* is set to the packet length; *pkt_len* is set to the packet length, too; *appl_id* is set to the ACB identifier.	Some bit fields in the *b_sh* member are initialised, notably: *type* is set to SH_BULK or SH_INTER, according to the type of the data packet; *rounding* is set to the two least significant bits of the negative of the packet length; *LAST* is set to 1; *frag_num* is set to 1.
The buffer is enqueued in the *a_bufq* queue of the relevant ACB.	The buffer is enqueued in the *bulk_q* or in the *inter_q* queue, according to the type of the data packet.

The action of the *LAN isr* is done.

Now it is the turn of the **Drea task** to use the buffer fields in order to build the control sub-burst and the transmit program. The *Drea task* does not modify anything in the buffer chains structure, so no contention problem with other tasks exists at the buffer level.
Indeed, given that the buffer chains are singly linked, the buffers can be enqueued and dequeued without the *Drea task* is ever aware of this.

Stream data: Drea task actions	Datagram data: Drea task actions
The buffer *b_csw* member is used to initialise the *csw* of the relevant data sub-burst. Anyway, the *cswl* and *rate* bit fields are set according to the current values contained in the relevant ACB.	The buffer *b_csw* member is used to initialise the *csw* of the relevant data sub burst.
The buffer *b_sh* member is used to initialise the satellite header of the relevant data sub-burst.	The buffer *b_sh* member is used to initialise the satellite header of the relevant data sub-burst.
If the data contained in the current buffer do not entirely fit in the data sub-burst, the *frag_len* bit field is decremented in the *b_sh* member and the *b_datap* member is decremented by the amount of data that fits in the data sub-burst. The Drea task then stops processing the buffers in this ACB buffer chain. The processing of this ACB buffer chain will begin from the current buffer in the next frame.	If the data contained in the current buffer do not entirely fit in the data sub-burst, the *frag_num* bit field in the *b_sh* member is incremented by one. The *cswl* bit field in the *b_csw* member, the *rounding* bit field in the *b_sh* member and the *b_datap* member are then set accordingly to the length of the data yet to be sent from this buffer. The PDB then stops processing the buffers in the datagram buffer chain. The processing will begin from this buffer in the next frame.

The buffers are now used by the **FHF isr**.
This interrupt routine pushes the data contained in the buffers into the Tx FIFO. When finished, if the relevant buffer has no more data in it to be transmitted in the next frame, the B_SENT flag is raised and the count in the relevant buffer header is

updated. This means that the count member in the ACB buffer chains and in the two datagram buffer chains contains the number of buffers get to be processed , not the number of enqueued buffers.
The buffers marked as *sent* wait now for the **Clrq** task to be awakened (once every frame, but it has quite a low priority) and to release them. The Clrq task has the same mask as the LAN isr interrupt level. Since a task cannot interrupt an isr routine, the Clrq task and the LAN isr have mutually exclusive access to the buffer queues. Since no other object manipulates the buffer queues, no contention problem arises. The FHF isr, on the other hand, accesses the queues only for reading, as previously stated.

SECTION 6. AREAS PRESENT IN THE *DOWN* PROCESS ONLY

6.1 The Station Control Block (SCB)

The parameters of each station are contained in an associated Station Control Block (SCB) whose format is in Fig. 13.

byte	length	
4	pointer to next SCB in the datagram circular list	s_next
4	pointer to previous SCB in the datagram circular list	s_prev
4	station active (flag)	S_ACTIVE
4	datagram request enqueued in the DRQ (flag)	S_DRQ
4	station in the list to be skipped (flag)	S_SKIP
4	stream request pending (flag)	S_STREQ
4	fade level has changed (flag) (same flag defined in 1.1.1)	S_FADECHANGED
4	stream request is extra (flag)	S_EXTRA_STREQ
4	station can become the new master in case of master fault (flag)	IS_POSSIBLE_MASTER
4	station in starting phase (flag)	S_STARTING
2	datagram request (in 16 words)	s_dtgreq (1)
2	datagram assignment (in channel words)	s_dtgass (1)
4	datagram request in channel words	s_dtg_cw
2	stream request (in words per frame)	s_streq (1)
2	stream assignment in words per frame	s_stass (1)

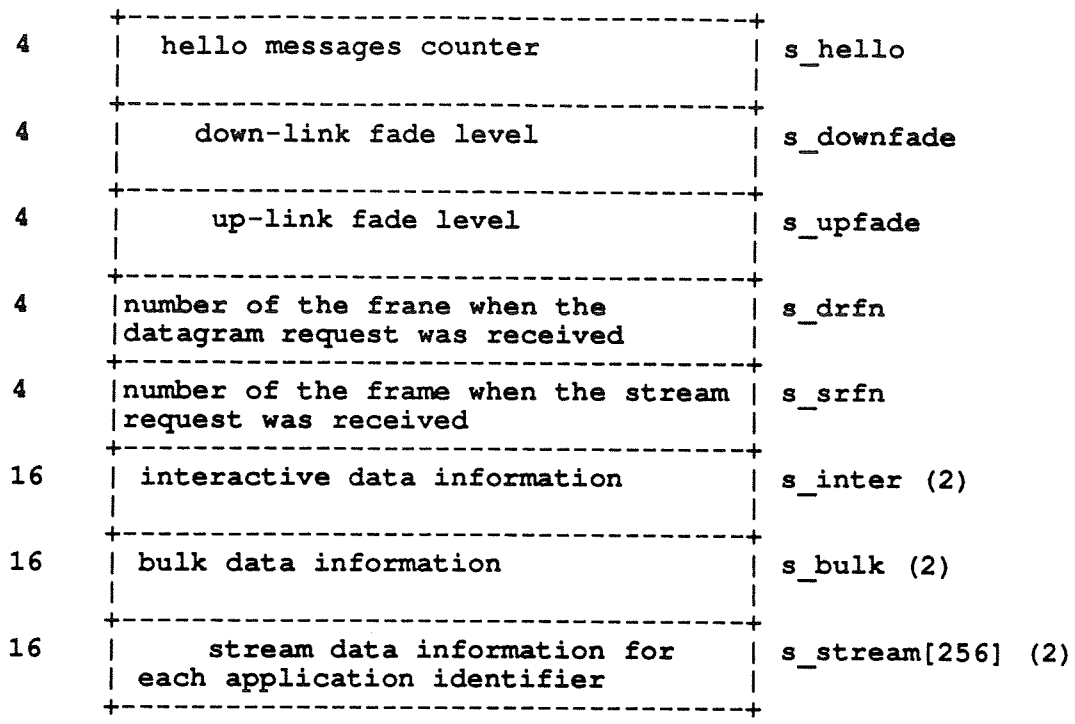


Fig. 13. The Station Control Block (SCB)

(1) These members have the same format shown in 1.1.1.2 for the requests in the cca area (here repeated in Fig. 13.1):

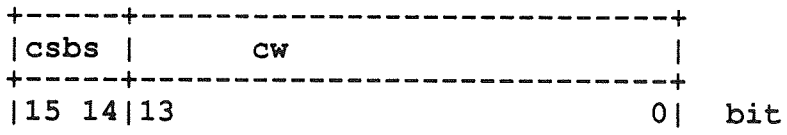


Fig. 13.1 The *s_dtgreq* and *s_streq* sub-fields format

csbs 2 bits. Control sub-burst speed. Minimum bit rate used for transmitting the CSB. It is computed according to the algorithm described in the Introduction chapter for the datagram request.

cw 14 bits, Number of requested channel words.
The stream request is expressed in words per frame (maximum stream throughput which can be expressed in 14 bits = 25.6 Mbit/s when the transmission bit rate is 8 Mbit/s).
The datagram request is the DGR as defined in the Miscellaneous Concepts section of the Introducion chapter.

(2) The format of each of these members is shown in Fig. 13.2.

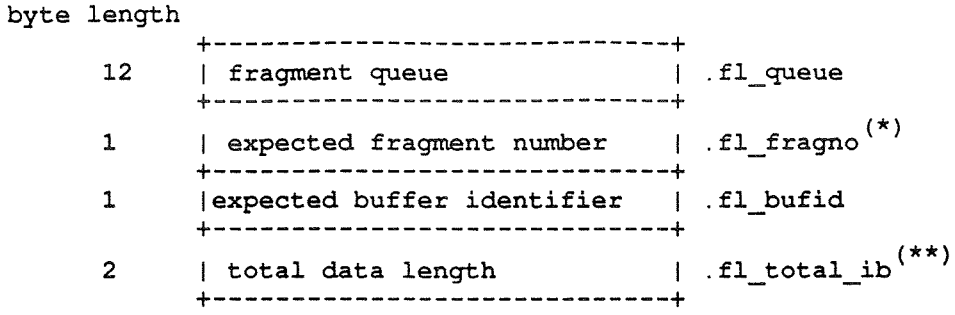


Fig. 13.2. The "information" member

(*) used by datagram only.
(**) used by stream only.

The fragment queue is a queue of dn_buffers, structured as shown in 13.3:

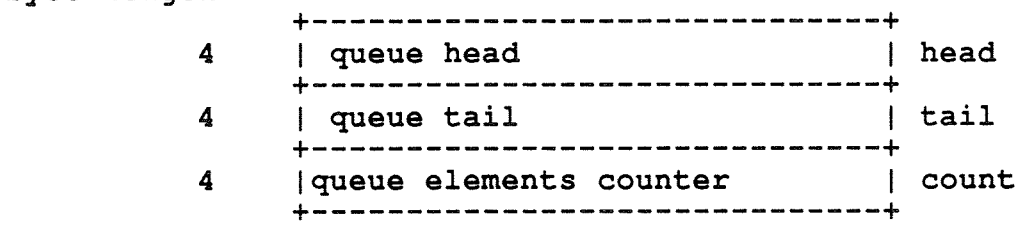


Fig. 13.3. The "fragment queue"

6.2 The Down Statistic area (down_statistic)

It is described in Fig. 15. Each field is a counter.

byte length		
4	FHE interrupts count	fhe_count
4	EOF interrupts count	eof_count
4	SOF interrupts count	sof_count
4	RBR interrupts count	rbr_count
4	times FHE rewinds the counter	fhe_rewind_count
4	times fheclean has found something wrong	fhe_cleanup
	times readrb trapped to fheint	readrb_trap_count
4	RB UW miss	rb_uw_miss
4	RB CRC wrong	rb_crc_wrong
4	RB correctly received	rb_good
4	control sub-burst UW miss	csb_uw_miss
4	control sub-burst CRC wrong	csb_crc_wrong
4	control sub-bursts correctly received	csb_good
4	total number of received DSB even if not addressed to me	dsb_total
4	stream DSBs UW misses	sdsb_uw_miss
4	datagram DSBs UW misses	ddsb_uw_miss

4	stream DSBs received with wrong CRC	sdsb_crc_wrong
4	interactive DSBs received with wrong CRC	idsb_crc_wrong
4	bulk DSBs received with wrong CRC	bdsb_crc_wrong
4	stream DSBs received without errors	sdsb_good
4	interactive DSBs received without errors	idsb_good
4	bulk DSBs received without errors	bdsb_good
4	received packets with unknown code from the LAN	lan_wrong_packets
4	fatal errors in Rx from the LAN	lan_fatal_err
4	missed packets in Rx from the LAN	lan_missed_err
4	LAN Rx framing errors	lan_fram_err
4	total received packets from the LAN	lan_rx_total
4	CRC errors in Rx from the LAN	lan_crc_err
4	total packets transmitted to the LAN	lan_tx_total
4	total packets transmitted with errors to the LAN	lan_tx_err
4	number of packets whose Tx to the LAN was deferred	lan_deferred
4	number of packets sent at first try	lan_one_try
4	number of packets not sent at first try	lan_more_try

Fig. 15 The *down_statistic* area format

FODA/IBEA-TDMA. Implementation specs.

6.3 The queues in the Down Process

6.3.1 The LANCE queues

Free buffers are enqueued in the **free_buffers** queue.

Buffers addressed to the LAN and containing bulk data are enqueued to the LANCE **lanceq[BULK_Q]** queue.

Buffers addressed to the LAN and containing interactive data are enqueued to the LANCE **lanceq[INT_Q]** queue.

Buffers addressed to the LAN and containing stream data are enqueued to the LANCE **lanceq[STREAM_Q]** queue.

6.3.2 The SCB queues

SCBs having a pending datagram request are enqueued in a datagram circular list. An SCB passing *from datagram request zero to datagram request n,* is put at the top of the circular list.

SECTION 7. SOME BEHAVIOURS IN THE DOWN PROCESS

7.1 The reception from satellite

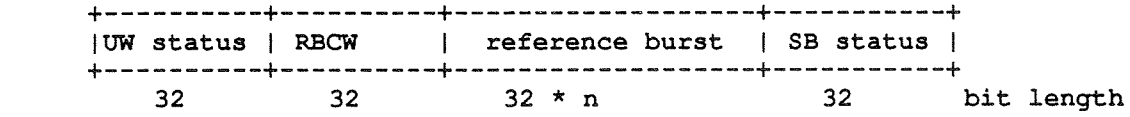
Data received from satellite are put by the hardware in a FIFO 128 words long.
The FIFO must be emptied by the software. Then data must be analysed and the original data packets reconstructed and passed to the LAN.

The FHE isr empties the FIFO. It is written in assembler.
Its functions are:

1) When an RB is received, write it in the rb_got member of the communication CA area (see section 3).
2) When a CSB is received, read the length of the CSB and how many DSB are presents.
3) For each DSB, read the satellite header in order to verify if it is a datagram or a stream sub-burst.
4) If it is a datagram sub-burst, get a buffer, read the data in the buffer and enqueue it in the CSB_AREA.
5) If it is a stream sub-burst, more than one buffer may be filled with the data. The filled buffers must be enqueued in the CSB_AREA for further analysis.
   For both the stream and the datagram sub-bursts, buffers contain data fragments.

After the assembler software step, data to be analysed have one of the two following formats:

1) REFERENCE BURST as *rb_got* member of the CA area (see section 3)



2) CSBs and relative sub-bursts

A circular list of chunks of memory (named CSB_AREA) is used for the reception of the CSBs.
The assembler software uses the *the_csbp* pointer to point to the current chunk where the CSB is put (Fig. 16).
The format of the data moved in each chunk is shown in Fig. 17.
Each chunk in the circular list is associated with a received data burst and contains information related to the received data burst (*burst descriptor* field).

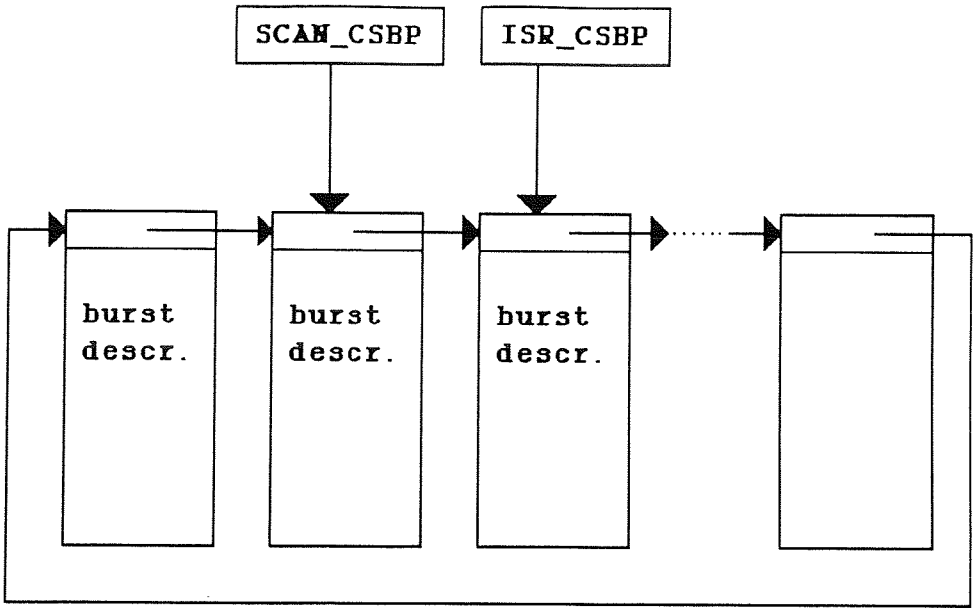


Fig. 16. The CSB_AREA

The *format of each dn_buffer in the circular list of the CSB_AREA* is described in Fig. 17.

byte length		
4	pointer to next descriptor	bd_next
4	(CSB) UW status	bd_uw_stat
4	ccsw of the burst	bd_ccsw
4095 * 4	Control sub-burst (max length)	bd_body
4	TOVH	tovh
4	(CSB) status	bd_cstat
12	data sub-burst N.1 descriptor block	bd_sb[1]
12	data sub-burst N.2 descriptor block	bd_sb[2]
	. . .	
12	data sub-burst N.MAX_CSW descriptor block	bd_sb[MAX_CSW]

Fig. 17. dn_buffer format in the circular list of the CCA_AREA

(*) as long as MAX_CCSW * 4 + cca length _MAX_CSW * 5

where:

(CSB) UW status relevant to the CBS UW detection. See format in note 1 of section 3.

CSB control sub-burst. See format in section 1. The space devoted to it in a chunk is fixed and equal to the maximum possible (byte) length of a control sub-burst.

CSB status it is a status relative to the CSB. See format in note 1 of section 3.

The descriptor block of the data sub-burst #i has the format shown in Fig. 17.1.

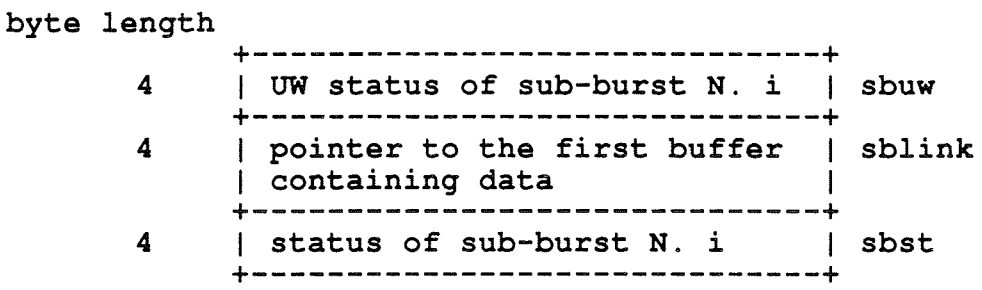


Fig. 17.1 - The descriptor block of a sub-burst

where:

sbuw relevant to the SB #i UW detection. See format in note 1 of section 3.

sblink pointer to the first of a list of buffers containing the data sub-bursts #i. The data buffers are filled starting from the b_data offset (see Fig. 5-B of section 2). On the top of the DSB data, the GAFO header may be added (if first buffer of a packet). Each buffer contains a fragment.

sbst status relative to the sub-burst #i. See format in note 1 of section 3.

7.2 When the master DOWN process receives a stream request from a slave station

A slave can request to the master either a Normal stream request or an Extra stream request. When a normal request is received, it is satisfied if the SSF (that is the total frame space occupied by the stream traffic) will not exceed the NSUB (Normal Stream Upper Bound) The request is not entirely satisfied if the NSUB will be exceeded. If the application that caused the request is compressible and can fit in the reduced assignment, it is requested to compress itself. If not, the slave station releases the insufficient stream space and refuses the application request.

When an Extra stream request is received, it is satisfied if the SSF will not exceed the ESUB boundary. The request is not entirely satisfied if the ESUB boundary will be exceeded. The slave station distributes the received (extra) allocation among the suffering stations, i.e. among those applications that need more bandwidth in order to maintain their class of service.

7.3 Stream/datagram requests and their assignments

The **stream request** in expressed as a multiple of the word per frame. The **datagram request** is defined in the Introduction chapter. Both the stream and the datagram assignments are expressed in words.

The station assembles all the stream requests received from the various applications, adds the burst overhead and sends to the master station one unique global stream request.
The most significant two bits of the stream request issued to the master indicate the minimum bit rate the station needs for transmitting the CSB of the stream data burst. **Each stream request (and relevant assignment) is considered "per frame" until the master receives an explicit relinquish command.** In the frame, the datagram slot, if any, always starts immediately after the stream slot.

A slave station distributes the received assignment among the requesting applications. The master uses an *ad hoc* algorithm for the distribution of the datagram space among the requesting stations. The stream space is distributed in FIFO order among the requesting stations. This is quite different from the datagram assignment technique, as the latter is performed on a frame by frame basis, while the stream assignment to a station is not modified until explicitly requested by the station itself.

The transmission window is computed as sum of the stream and of the datagram assignments, if the preamble bit rate of the stream assignment is not greater than the preamble bit rate of the datagram assignment. If it is greater, two consecutive slots must be set up, to not penalize the stream assignment by using a larger (unique) preamble due to the worst transmission conditions of the datagram data.

7.4 The master station fault recovery

All the stations can play the role of master.
Each slave station which can become master in case of fault of the current master station sets ON the IS_POSSIBLE_MASTER bit in the control flags field of the CCA area. The master station records this condition by setting ON an appropriate bit in the *status* field of the SCB of the relevant station. By scanning the SCBs, the address of the first station available to become the new master is reported in the reference burst. If a slave decides that the role of master cannot more be supported (station in fade) it sets OFF the IS_POSSIBLE_MASTER bit in the CCA area of the CSB. The master station again records this condition in the status byte of the relevant SCB. If the renouncing station was the first one in the SCB list, the master reports in the reference burst the address of the possible next master station.

REFERENCES

[1] R. Beltrame, A.B. Bonito, N. Celandroni, E. Ferro: "FODA/IBEA-TDMA: final report on the new protocol for mixed traffic. Theoretical study and first simulation results", CNUCE Report C85-03.

[2] N. Celandroni, E. Ferro: "FODA/IBEA-TDMA satellite access scheme for mixed traffic. Implementation and testing features at 2Mbit/s bit rate. Final Report.", CNUCE Report C88-10, April 1987.

[3] R. Beltrame, N. Celandroni: "The performances of the FODA/IBEA access scheme: theory and simulation results", CNUCE Report C86-18.

[4] Celandroni N., Ferro E., Marzoli A.: "Fade detection in the FODA/IBEA system", proceedings of the Olympus Utilization Conference, Vienna (A), 12-14 April 1988.

[5] Celandroni N., Ferro E.: "The FODA/IBEA-TDMA satellite access scheme: presentation, study of the system and results", IEEE Transaction on Communication Magazine, Vol. 39, N. 12, pp. 1823-11831.

[6] Celandroni N., Ferro E.: "A deterministic study of the FODA/IBEA frame utilization at variable bit and coding rates", CNUCE Report C89-24, October 1988.

[7] Celandroni N., Ferro E.: "Protocol between the FODA/IBEA system and the outside environment. The GA-FO protocol.", CNUCE Report C91-21, 1991.

[8] GEC-MARCONI Research Centre:
"Working specification for Olympus TDMA equipment", Issue 2.0, Y/212/9883, August 1990.

[9] GEC-MARCONI Research Centre:
"Olympus TDMA equipment. Software Manual. Issue 2", MTR 91/60A/, Y/212/10441, September 1991.

[10] GEC-MARCONI Research Centre:
"Olympus TDMA equipment. User Manual. Issue 2", MTR 91/60A, Y/212/10440, September 1991.

[11] GEC-MARCONI Research Centre:
"Olympus TDMA equipment. Transmit and receive controllers. Hardware manual. Issue 2.", MTR 91/60A, Y/212/10442, Sempember 1991.

[12] GEC-MARCONI Research Centre:
"Olympus TDMA equipment. Burst mode modem equipment manual", MTR 91/60A, Y/212/10443, Sempember 1991.

[13] Ferro E.: "A satellite network for good weather conditions and for high rain attenuation", proceedings of the SBT/IEE International Telecommunication Symposium, Rio de Janeiro, Brazil, September 4-6, 1990.

[14] Ferro E., Celandroni N.: "FODA/IBEA-TDMA. Satellite access scheme for mixed traffic at variable bit and coding rates. System description", CNUCE Report C92-05, April 1992.

Appendix A

IMPLEMENTATION DETAILS ON THE UP PROCESS

A.1. The timer management

The Timer functions provide FODA/IBEA with the possibility to send a message after a certain period of time is expired.
A timer dependent message is sent to the UPlh task to notify a timer expiration.

---

start_timer(type, acbp)

arguments: timer type (suffer, request or ACK) and address of the application's ACBs in the case of ACK timer.

return: none.

description: for the suffer and request timers it initialises the time member of the appropriate timer control block (TCB). If an ACK timer is activated, the time counter member is initialised and the ACB is enqueued at the end of the *timer_wsrq* queue. Given the limited number of contemporaneously active applications and the low probability of having a lot of ACK timers active at the same time, the timer queue is not ordinate.

---

stop_timer(type, acbp)

arguments: timer type (suffer, request or ACK) and address of the application's ACB in the case of ACK timer.

return: none.

description: used to stop an active timer. In the case of suffer or timer request, the timer value is initialised to -1. In the case of ACK timer, on the base of the appropriate input address, the relevant ACB is dequeued from the timer queue.

---

time_tick()

arguments: none

return: none

description: called once every frame. It decrements the time fields of all the active timers. If the time value reaches zero, an appropriate message

is send to the UPlh task, and, in the case of ACK timer, the ACB is dequeued from the timer queue.

A.2 Global variables

int *R_STREAM_REQUEST_READY*

Boolean variable. When TRUE, the *prepare data burst* (PDB) interrupt routine must send the total stream channel request to the master. Set on by the UPlh task and off by the PDB interrupt routine.

int *R_NSUB_EXCEEDED*

Boolean variable. When TRUE, new applications cannot be accepted because the stream sub-frame boundary NSUB has been reached. Set on/off by the Drea task, and read by the UPlh task.

int *R_ESUB_EXCEEDED*

Boolean variable. When TRUE, all stream channels are distributed among stations. It is possible to reach this limit only in situations of fade. Set on/off by the Drea task, and read by the UPlh task.

int *R_EXTRA*

Boolean variable. When TRUE, the stream channel allocation request which has to be sent is due to changes of already active applications. Set on/off by the UPlh task and read by the Drea task.

request (2 + 14 bits) *r_updated*

Total number of stream channels and preamble bit rate needed by the station. Written by the UPlh task, read by the Drea task

request *acb_current.req*

Part of the ACB. It contains the number of stream channels and the preamble bit rate granted to the application by the master station. Written by the UPlh task, read by the Drea task.

byte *acb_current.code*

Part of the ACB. It contains the current coding rate for the data of this application. Written by the UPlh task and read by the Drea task.

byte *acb_current.speed*

Part of the ACB. It contains the current transmission speed of the data of this application. Written by the UPlh task and read by the Drea task.

request *r_granted.noc*

Number of stream channels and preamble bit rate granted to the station by the master. Written by the Drea task, read by the UPlh task and by the Drea task.

A.3. The UP LAN Handler task

The **UP LAN Handler task** (**UPlh**) handles various events coming from the LAN and from the master station. It handles the timer expirations too. Information about events are presented by using the VMEexec message queue services. This task is designed as an endless loop. If no events have to be handled, the task sleeps on the message queue.
Each message represents one event to the program. Program sends cumulative requests for stream channel to the master, which means that stream channels requests are sent when there are no messages in the queue. Two types of requests are distinguished, one due to fade changes (Extra requests) and another one due to new applications (Normal requests). Extra requests have higher priority.

Possible events are:

APPLICATION_STREAM_REQUEST

APPLICATION_STREAM_RELINQUISH

SUFFER_TIMER_EXPIRED

ACK_FOR_MODIFICATION

ACK_TIMER_EXPIRED

REQUEST_TIMER_EXPIRED

OTHER_STATION_FADE_CHANGED

STREAM_ALLOCATION_CHANGED

CONGESTION_DETECTED

CONGESTION_OVERCOME

A.3.1 Overview of the used routines

acb ****appl_stream_request (msg)***

arguments: address of the event message.

return: address of the ACB allocated to the new application, or NULL when there are no more new applications.

description: allocate a new ACB for the application, initialise the ACB fields, initialise the transmit parameters and put the ACB in the *new_wsrq* queue.

---

void appl_stream_relinquish(ACB)

arguments: address of the application control block

return: none.

description: dequeues the application's ACB from all the queues where it was enqueued and deallocates the ACB.

---

void compute_extra_request()

arguments: none.

return: none.

description: it scans all the active applications and computes the total number of stream channels needed for this station, builds the extra queue and updates the r_updated, r_next_csb and r_next_dsbovh values. It is never called when there is an outstanding request due to the new applications.

---

void compute_new_request()

arguments: none

return: none

description: it scans the applications waiting in the areq_queue, computes the total number of stream channels needed for this station and updates the r_updated, r_next_csb and r_next_dsbovh values.

---

int fade_change()

arguments: none

return: Boolean value indicating if fade change influences our application.

description: look for all applications which send data to the stations which have changed their fade level and, if necessary, update the ACB's transmission parameters according to the new fade levels. It also updates the transmission fade for all user groups.

---

int distribute_extra_stream_channels

arguments: none

return: FALSE if all channels are distributed, TRUE if some channels remain unused.

description: it distributes stream channels among the existing applications which have to counter a fade level. If there are not enough channels to

satisfy all the applications in the extra_wsrq queue, for each application the function:

- satisfies the applications which need less channels than before,
- tries to satisfy the necessary redundancy. If not possible,
- compresses the application (if possible). Some applications may eventually work out of specifications.

On entry, the function expects the extra queue already built and the number of allocated stream channels and valid preamble bit rate (pbr) already set in the r_granted member.

---

int ***examine_healthy_applications(gained)***

arguments: number of gained channels in the last master's reply.

return: updated number of gained channels.

description: it looks for applications in the extra queue which can be satisfied. It satisfies them and dequeues them from the extra queue. It updates the gained number of stream channels.

---

int ***examine_csb(gained)***

arguments: number of currently gained channels.

return: updated number of gained channels.

description: it tries to satisfy the change in the control sub-burst size is made. If it is not possible to use a correct pbr, an attempt is made to improve it.

---

int ***examine_suffering_applications(gained)***

arguments: number of currently gained channels.

return: number of channels which have to be returned to the master.

description: it tries to improve the preamble bit rate, if necessary, and then to satisfy the suffering applications. If the gained number of channels is not enough to satisfy all the suffering applications, some of them will work out of specification. It destroys the extra queue.

---

int ***distribute_new_stream_channels()***

arguments: none

return: Boolean value indicating the necessity to return the unused channels.

description: it distributes stream channels among the new applications. If there are not enough channels to satisfy all the new applications, unsatisfied applications are refused. On entry, it expects the new applications queue to be already built and the number of allocated stream channels and valid pbr already set in the r_granted member.

---

hword ***compute_csb(which)***

arguments: csb overhead to be computed considering current or next transmission situation.

return: Control sub-burst occupation expressed as a number of stream channels.

description: it calculates the number of channels necessary for the transmission of the control sub-burst. It updates the *r_next_csb* or *r_current_csb* member.

---

hword ***overhead(csb_param, appl_param)***

arguments: parameters specifying which CSB, pbr and which application's speed must be taken into consideration in order to calculate the r_current_dsbovh or r_next_dsbovh member.

return: Total number of channels needed for r_current_dsbovh or r_next_dsbovh.

description: it calculates the number of channels needed for data sub-burst preambles in the current and next situation.

Appendix B

IMPLEMENTATION DETAILS ON THE DOWN PROCESS

B.1. The buffer management

The buffer structure in the down-side software differs from the up-side buffer structure. Nevertheless functions for buffer management are the same as in the UP side :

init_buffers()	-buffer space initialisation
get_buffer()	-allocate free buffer and get its address
release_buffer(addr)	-release used buffer.
release_packet(addr)	-release all the buffers of a packet.

The function *get_buffer* is called from the FHE (Fifo Half Emptyl) interrupt routine, when a data sub-burst is received. The function *release_buffer* is called from the DNLH (Down LAN Handler) task or from the LAN interrupt routine. The FHE interrupt has a higher priority than any other interrupt, so the *release_buffer* operation has to be masked at the FHE interrupt priority.

B.2. The queue management

Structures in which buffers, packets or control blocks are linked are divided in queues and circular lists. In the same way, the functions which manipulate these queues or lists are divided.

Queue manipulation functions are:

insert_buffer_at_queue_tail()	inserts a buffer at the tail of a queue and increments the queue counter;
insert_packet_at_queue_tail()	inserts a packet at the tail of a queue and increments the queue counter,
remove_packet_from_queue_head()	removes a packet from the head of a queue and decrements the queue counter.

List manipulation functions are:

insert_scb_in_list ()	inserts a station control block in a circular list and increments the list counter;
remove_scb_from_list ()	removes a station control block from a circular list and decrements the list counter.

B.3. Global Variables

struct scb ***cur_scbp***

Pointer to the first scb in the datagram circular list. Referenced by the DNlh task and by the timer interrupt routine (in the *analyse_datagram_request* function).

struct rb_list_entry ***young_rb***

Pointer to the last received reference burst in the RB circular list. Referenced by the timer interrupt routine and to prepare the receive program.

struct scb ***scb[MAX_ST+1]***

Station control blocks array. First MIN_ST - 1 entries are empty (they correspond to the user groups in the UP side). Referenced in most part of the software.

struct burst_desc ***burst_desc[BD_LIST_SIZE]***

Circular list of the data burst descriptors. Written by the FHE isr, read by the DNlh task.

struct burst_desc ***scan_csbp***

Scanning burst descriptor pointer. Updated by the DNlh task, tested by the FHE isr.

struct burst_desc ***isr_csbp***

Interrupt burst descriptor pointer. Updated by the FHE isr, tested by the DNlh task.

B.4. DOWN side tasks

Currently, the FODA/IBEA software on the Down side runs three tasks:

DNlh	down LAN handler task;
Uwrt	task which sends common data to the up side;
Prnt	common print error and warning messages task.

B.4.1. DNLH task

DNlh is the main task on the down side. It scans the circular list of the data burst descriptors analysing the received data bursts. The task is organized as an infinite loop waiting for the BURST_RECEIVED event. This event is sent by the FHE interrupt routine. The processing of the received data burst consists of checking its formal validity (unique word status, burst status), processing the received CCA field and reassembling the data packets.
The processing of the received CCA entails the update of the station state variables (fades, flags) and the registration of the station stream and datagram requests.
Incomplete stream packets are sent to the LAN, while incomplete datagram packets are thrown away.

When all the received data burst descriptors are processed, the task suspends itself waiting for a new event from the FHE interrupt routine.

B.4.1.1 LAN interface

The LAN interface consists of the *send_packet (packp, p_type)* function where *packp* is the pointer to the packet's first buffer, and *p_type* is the packet type (STREAM, INTERACTIVE or BULK). The function finds out the packet size and the number of fragments. It tries to reduce the number of fragments if they are too short (shorter than MIN_FRAG_SIZE), by copying them in the preceding fragment. If the packet has more than one one fragment, the function fills the first fragment until a minimum length (100 bytes) is reached This is imposed by the LANCE chip. After the Ethernet header is built, the packet is inserted at the tail of one of the three LANCE queues used for stream, bulk and interactive data, respectively. Then, the initialization to send data is performed. The initialization is priority ordered, with stream queue with highest priority and bulk queue with the lowest. The unsent packets remain in the queues, waiting for LANCE interrupts which repeat the transmit initialization untill there are packets to be sent.

B.4.2. Uwrt task

The Uwrt task has the simple job of sending to the UP process a data array (common to the UP and to the DOWN processes) by using the COM interface (currently the MARCONI SCSI bus driver). The task is implemented as an infinite loop. It waits for the START_SENDING_CA event issued by the timer interrupt routine. After the event is received, the task sends the common data area and suspends itself waiting for another START_SENDING_CA event.

B.4.3. Prnt task

The Prnt task is a temporary solution for printing debug messages. It is supposed that this task will not be present in the final version of the FODA/IBEA software, at least not in the current form. It is implemented in order to print messages from the interrupt routines avoiding the interference of messages from different tasks.

B.5. Interrupt service routines

Currently, in the FODA/IBEA software of the DOWN side, the RBR (Refernce Burst Received) isr, the FHE isr and the LAN isr are implemented. The FHE isr is implemented in assembler, the other two in C language.

B.5.1 FHE interrupt service routine

This isr handles the reception of the data from the FIFO and puts the received data in buffers enqueued to the burt desriptors. It takes care of the management of the UW misses, of the paersing of the CSB and of the demultiplexing of stream DSBs. The stream DSB can contain a mix of whole and of fragmented packets. Each fragment is put in a separated buffer.

B.5.2. RBR interrupt service routine

The RBR isr is enterd 100 microseconds after the RB has been entirely received. The routine analyses the received reference burst, prepares the next reference burst to send, initiates the transmission to the UP process of the common data area, checks the stations activity state and prepares the event memory.

If the received reference burst is error free, it is copied in the circular list of the last 14 received reference bursts. If the master station is going down, the master change procedure is initiated.

The reference burst to transmit in the next frame is prepared in the common CA area. During its preparation, all the received stream and datagram requests recorded in the DNlh task are analysed. Differential sum of the allocated space in the frame is checked calculating the total frame space allocation.

The transmission of the common CA area from the DOWN to the UP side is initiated by sending the START_SENDING_CA event to the Uwrt task.

Every CHECK_ALIVE_MASK number of frames it is necessary to check if the active stations have had some activity meanwhile. If no activity is noticed (nothing at all is sent from a station), the station is declared *dead* in the reference burst by using the DEAD_FADE value as down-fade level of that station.

B.5.3. LAN interrupt service routine

At the writing time, the LANCE transmit interrupt service routine only has been implemented. The receive routine will be implemented. After checking the statuses, the number of busy transmit descriptors is decremented and the buffer is released. At the routine end an attempt is done to send another packet.