EBU Technical Recommendation R42-1997
Adoption of a protocol for the exchange of teletext pages

EBU Committee	*First issued*	*Revised*	*Re-issued*
PMC	1986		1997

Keywords: Teletext

The EBU *considering*

- that services using data broadcasting techniques - such as teletext and subtitling - have already been introduced by most of the Members;
- that it would be beneficial if the Members were able to exchange information coded in teletext form;
- that the public data-communications networks constitute an efficient support for such exchanges;
- that two different teletext systems are in use among the Active Members;

recommends

the application of the protocol specified in the *Appendix* to this Recommendation for the exchange of teletext pages between the EBU Members.

Appendix
Specification of a protocol for the exchange of teletext pages

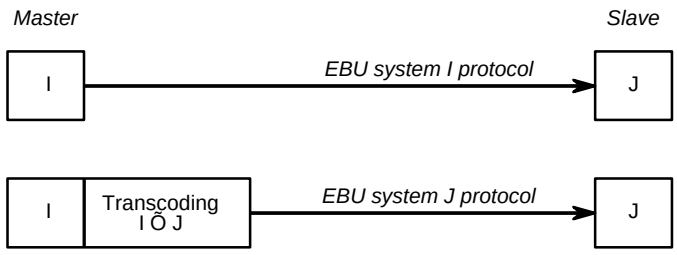
1. Basic principles

This specification describes the procedure for the exchange of pages between two teletext editing centres.

An intelligent interface is needed to make data communication possible between the various editing systems. This interface could be either a microcomputer or the host computer of the system. For the purpose of simplification, only one exchange protocol has been defined for fixed-format teletext and another one for variable-format teletext systems. A complete exchange between all EBU Members implies the implementation of the transcoding facility between data of the fixed-format and the variable-format systems. For the exchange of pages, the following rules apply:

- each broadcaster who wishes to offer pages to others must implement the EBU protocol corresponding to his teletext system;
- each broadcaster who asks for a page exchange and initiates the exchange - the *master* - must implement the EBU communication protocol of his partner who acts as *slave*;
- if transcoding is necessary, this transcoding must be done by the master.

These basic principles are summarized in the following *figure*:



I and J represent different teletext systems.

The description of the procedure for the exchange of teletext pages is done in four different steps, i.e. at session level, at transfer command level, at page delivery level and at the transmission level. These steps are specified in the following sections.

2. Description of the session format

2.1. Repertory of session commands

`LOGIN <user identifier, (password)>`

This command opens a session between two systems which are able to exchange teletext pages. After sending this command, the system expects either a positive or a negative reaction from its partner. In the latter case, the reason may be specified in the returning message. Possible error messages can be: erroneous command, user unknown, password unknown, computer occupied, etc.

`LOGOUT <(user identifier)>`

This command closes a session between two systems which are able to exchange teletext pages. After sending this command, the system expects either a positive or a negative reaction from its partner. In the latter case, the reason may be specified in the returning message. Possible error messages can be: erroneous command, user unknown, etc.

2.2. Flow chart at session level

This flow chart is reproduced in *Fig. 1*. In this flow chart, rectangular boxes represent commands and rounded boxes represent states.

2.3. Variable-format exchange protocol at session level

a) A `LOGIN` command in variable-format is defined by the following byte sequence:

`S D X X X X X X Y Y Y Y ETX EOT`

with `X X X X X X` = the source identification
`Y Y Y Y` = the transmitter identification

A positive reaction to such a `LOGIN` command is represented by the byte sequence:

`S D Y Y Y Y X X X X X Z Z ETX EOT`

with `ZZ` = the number of the current session (in ASCII)

Possible negative reactions to this `LOGIN` command are:

`S D NAK` 33_{hex} `ETX EOT`
and `S D NAK` 39_{hex} `ETX EOT`

with 33_{hex} = `X X X X X X` and/or `Y Y Y Y` unknown
39_{hex} = erroneous command

b) A `LOGOUT` command in variable-format is defined by the following byte sequence:

`S F X X X X X X Y Y Y Y ETX EOT`

with `X X X X X X` = the source identification
`Y Y Y Y` = the transmitter identification

A positive reaction to such a `LOGOUT` command is represented by the byte sequence:

`S F ACK ETX EOT`

Possible negative reactions to this `LOGOUT` command are:

`S F NAK` 33_{hex} `ETX EOT`
and `A C NAK` 39_{hex} `ETX EOT`

with 33_{hex} = `X X X X X X` and/or `Y Y Y Y` unknown
39_{hex} = erroneous command

2.4. Fixed-format exchange protocol at sesslon level

For reasons of clarity, the description of this protocol is deferred to *Section 3.6.*

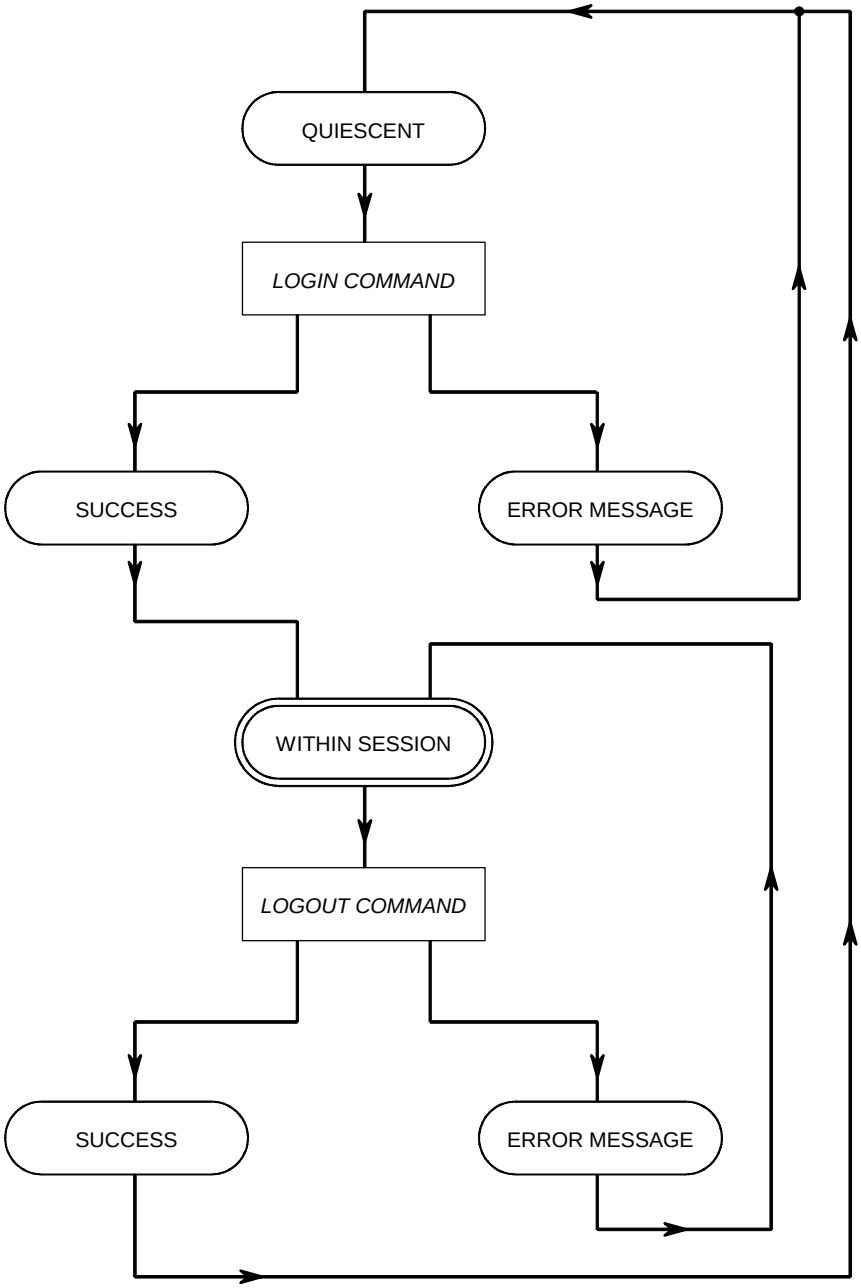


Fig. 1 - Flow chart at session level

In this flow chart, rectangular boxes represent commands and rounded boxes represent states.

3. Description of the transfer command format and of the page delivery protocol

3.1. Repertory of transfer commands

`READ PAGE (page number)`

This command is generated by the system which opened the session with a `LOGIN` command at session level, to invite its partner to send him the page indicated by the page number. In the fixed-format system, a page number is defined by up to 7 BCD digits (normally 3 BCD digits); in the variable-format system, a page number is defined by 3 BCD digits. After sending this command, the system expects either a positive or a negative reaction

from its partner. In the latter case, the reason may be specified in the returning message. Possible error messages can be: erroneous command, page number unknown, erroneous page number, etc.

`WRITE PAGE (page number)`

This command is generated by the system which opened the session with a `LOGIN` command at session level, to prepare its partner to receive the page characterized by the specified page number. In the fixed-format system, a page number is defined by up to 7 BCD digits (normally 3 BCD digits); in the variable-format system, a page number is defined by 3 BCD digits. After sending this command, the system expects either a positive or a negative reaction from its partner. In the latter case, the reason may be specified in the returning message. Possible error messages can be: erroneous command, erroneous page number, overload, etc.

3.2. Flow chart at transfer command level

This flow chart is reproduced in *Fig. 2*. In this flow chart, rectangular boxes represent commands and rounded boxes represent states.

3.3. Definition of a page in transmission

In the fixed-format system, a page is a series of data blocks of 44 bytes each, terminated by an extra `EOT` code. A page can contain up to 100 data blocks.

In the variable-format system, a page is a string of data bytes introduced by the control characters `SOH` and `RS` and ended by the control characters `ETX` and `EOT`. The string has a maximum length of 2048 bytes (the control characters included). The data bytes are odd.

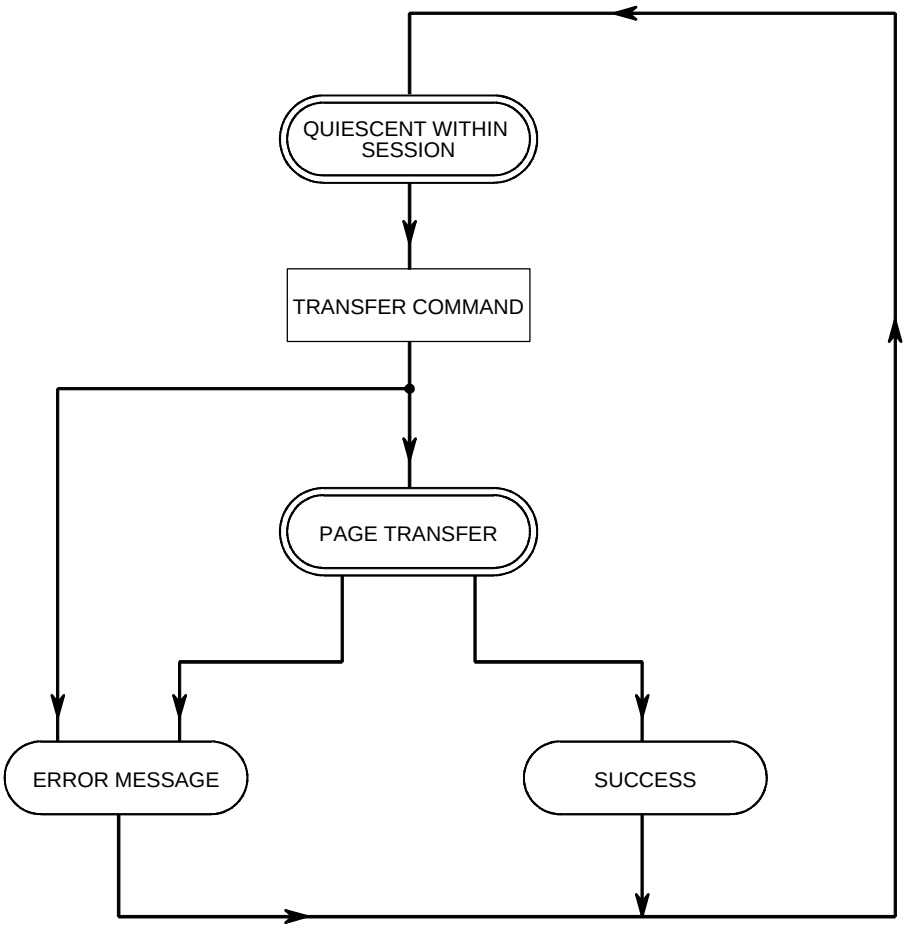


Fig. 2 - Flow chart at transfer command level

In this flow chart, rectangular boxes represent commands and rounded boxes represent states.

3.4. Transfer commands and page delivery protocol for variable-format systems

a) A READ PAGE command in variable-format is defined by the following byte sequence:

A C L D SOH RS Cl C2 C3 ETX EOT

with Cl C2 C3 = the Hamming coded page number.

A positive reaction to such a READ PAGE command is represented by the byte sequence:

A C Cl C2 C3 ACK A D SOH RS page ETX EOT

This reaction includes the transfer of the requested page.

Possible negatlve reactions to this READ PAGE command are:

A C Cl C2 C3 NAK 38_{hex} ETX EOT
A C Cl C2 C3 NAK 36_{hex} ETX EOT
and A C NAK 39_{hex} ETX EOT

with 36_{hex} = classification field erroneous
38_{hex} = page non-existent
39_{hex} = erroneous command

b) A WRITE PAGE command in variable-format is defined by the following byte sequence:

A C W D SOH RS page ETX EOT

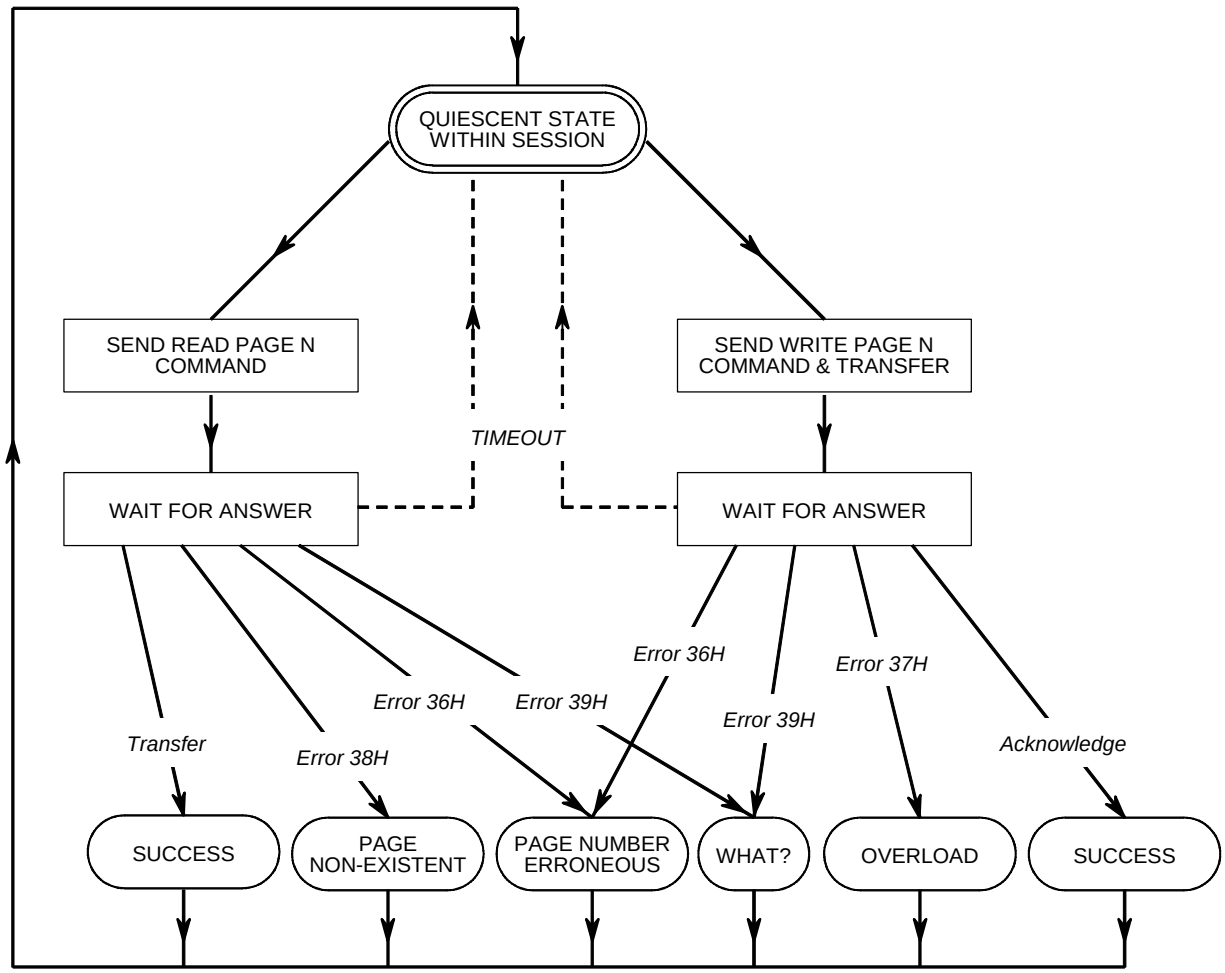


Fig. 3 - Flow chart for variable-format transfer commands and page delivery protocol

In this flow chart, rectangular boxes represent commands and rounded boxes represent states.

This command includes the transfer of the page.

A positive reaction to such a WRITE PAGE command (plus transfer) is represented by the byte sequence:

A C Cl C2 C3 ACK ETX EOT

Possible negative reactions to this WRITE PAGE command are:

A C Cl C2 C3 NAK 36_{hex} ETX EOT
A C Cl C2 C3 NAK 37_{hex} ETX EOT
and A C NAK 39_{hex} ETX EOT

with 36_{hex} = classification field erroneous
37_{hex} = transmitter capacity overload
39_{hex} = erroneous command

c) A flow chart illustrating the variable-format transfer commands and page delivery protocol is reproduced in *Fig. 3.*

3.5. Transfer commands and page delivery protocol for fixed-format systems

3.5.1. Row and character definition

The first character of a data row (R0 to R25, R26, R27 and R28) is SO ($0E_{hex}$), with odd parity. This is the first byte of a 44-byte SO block. The second byte is a seven-bit binary number uniquely representing a row, plus an odd parity bit. Data rows 24 and 25, where used, carry the binary numbering 30 and 31 for this purpose. The sequence continues with 40 bytes and ends with a 2-byte CRC code.

The first character of a command row (CRow) is SI ($0F_{hex}$), with odd parity. This is the first byte of a 44-byte SI block. The second byte is the seven-bit binary number 24 plus an odd parity bit. The sequence continues with 40 data bytes and ends with a 2-byte CRC code.

The 2-byte CRC code used to check a data row or a command row obeys generation and transmission rules similar to those applied for the basic page check word in packet 27, which checks the whole transmission of a page (see Section 12.3. and Fig. 8 of [1]).

Acknowledgement of a message is ACK (06_{hex}).

Rejection of a message is NAK (15_{hex}).

Clear screen (or page memory) command is FF ($0C_{hex}$).

Abort of page transfer by the master is ESC ($1B_{hex}$).

End of page transfer is EOT (04_{hex}).

All these latter characters are transmitted with odd parity.

3.5.2. Communication dialogues - general

In the *quiescent state within session*, the slave is waiting to receive a command row (CRow) from the master. The master is waiting for an input.

Leaving the quiescent state, the master sends a CRow with or without data rows following, in accordance with the ACK/NAK procedure. The slave receives the CRow and data rows if sent, and returns data rows if appropriate, also in accordance with the ACK/NAK procedure.

Note: The slave may transmit a CRow instead of ACK/NAK. This will terminate the dialogue and cause the *quiescent state within session* to be assumed.

If one system receives or transmits a large number of consecutive NAKs in a dialogue (say 10 consecutive NAKs), then the dialogue will be discontinued and the system will then return to its quiescent state.

A command abort function allows premature ending of long commands wrongly executed. When the initiative to abort comes from the master, it sends an ESC character to the slave. When the initiative to abort comes from the slave, it sends a CRow to the master; this CRow may indicate the reason for this abort action. An abort command must respect the dialogue turn.

3.5.3. Transmission of a page from the master to the slave; the WRITE PAGE N command

When a page has to be transmitted, the master sends a CRow and awaits a reply, which can be one of three responses:

a) The slave sends an ACK, which indicates to the master to continue with more data, i.e Rx, where x is first 00, then 27, 28, 26, 01, ..., 23, 24, 25 (and again 00, 27, 28, 26, 01, ..., 23, 24, 25 in case of pseudo pages).

Note 1: Up to fifteen packets 26's can be sent; each will have a different designation code.

Note 2: Only relevant rows need to be sent.

b) The slave sends a CRow, which indicates to the master to return to quiescent state after responding with an ACK.

c) The slave sends a NAK, which indicates to the master to repeat the previous transmission, i.e. the CRow sent by the master.

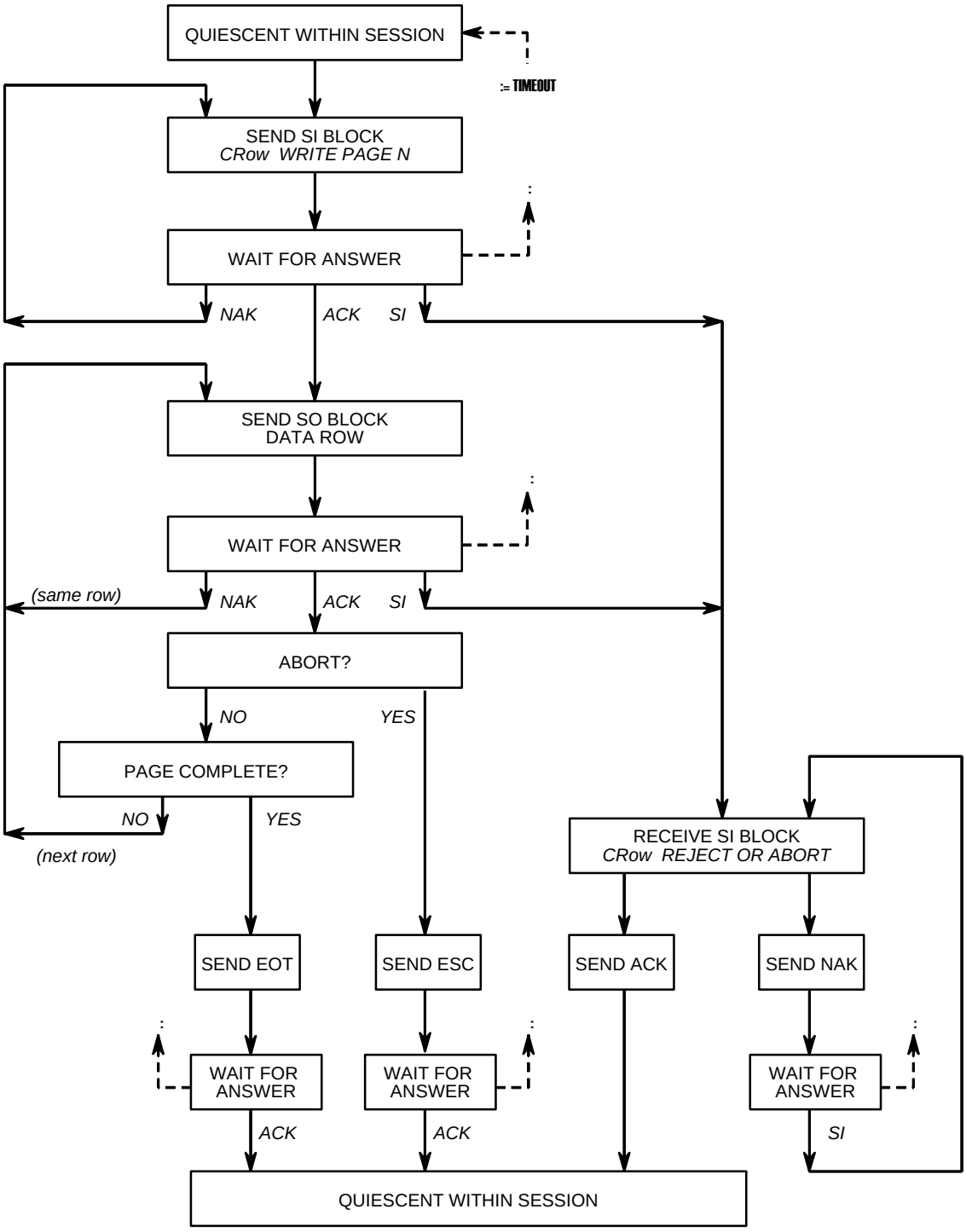


Fig. 4 - Fixed-format page delivery protocol in case of WRITE PAGE N transfer command

After having sent all relevant rows, the master indicates the end of a page transfer by transmitting an EOT character to the slave. If correctly received, the slave replies with an ACK, whereafter *quiescent state within session* will be assumed.

During the page delivery, the master may abort the transfer by sending an ESC character to the slave instead of a next data row. If correctly received, the slave replies with an ACK, whereafter *quiescent state within session* will be assumed.

The slave also has the opportunity to abort a page delivery process by sending a CROw to the master instead of an ACK or a NAK. This CROw may specify the reason for the abort action. The master can reply either with an ACK, whereafter *quiescent state within session* is assumed, or with a NAK, which implies the retransmission of the CROw by the slave.

This master-to-slave page transfer is reproduced in *Fig. 4*.

3.5.4. Transmission of a page from the slave to the master; the READ PAGE N command

When the master requests a page, it sends a CROw to the slave and awaits a reply which can be one of four responses:

a) The slave sends Rx which is stored by the master, and if these data have been correctly received, the master sends an ACK.

b) The slave sends a CROw which indicates to the master to return to quiescent state after responding with an ACK.

c) The slave sends a NAK, which indicates to the master to repeat the previous transmission, i.e. the CROw sent by the master.

d) The slave sends an FF character, telling the master to clear the screen (or the page memory), and will wait for an ACK to be returned after the master has cleared its display screen or page memory.

After having sent all relevant rows, the slave indicates the end of a page transfer by transmitting an EOT character to the master. The master must reply with an ACK, whereafter *quiescent state within session* will be assumed.

During the page delivery, the master may abort the transfer by sending an ESC character to the slave instead of an ACK or a NAK. If correctly received, the slave replies with an ACK, whereafter *quiescent state within session* will be assumed.

The slave also has the opportunity to abort a page delivery process by sending a CROw to the master instead of the next data row. This CROw may specify the reason for this abort action. The master can reply either with an ACK, whereafter *quiescent state within session* will be assumed, or with a NAK, which implies the retransmission of the CROw by the slave.

This slave-to-master page transfer is reproduced in *Fig. 5*.

3.5.5. Survey of possible command rows (CROw)

Depending on whether a page number is specified by 3 or 7 BCD digits, the WRITE PAGE command row has one of the two following formats:

SI / 24 / W N N N ———— / CRC

SI / 24 / W N N N S S S S ———— / CRC

Depending on whether a page number is specified by 3 or 7 BCD digits, the READ PAGE command row has one of the two following formats:

SI / 24 / R N N N ———— / CRC

SI / 24 / R N N N S S S S ———— / CRC

To indicate the rejection of a page transfer command or the abort of a page delivery by the slave, the command row is as follows:

SI / 24 / NN / ———————— / CRC

where NN is a number indicating the reason for rejection or abort, according to the following table:

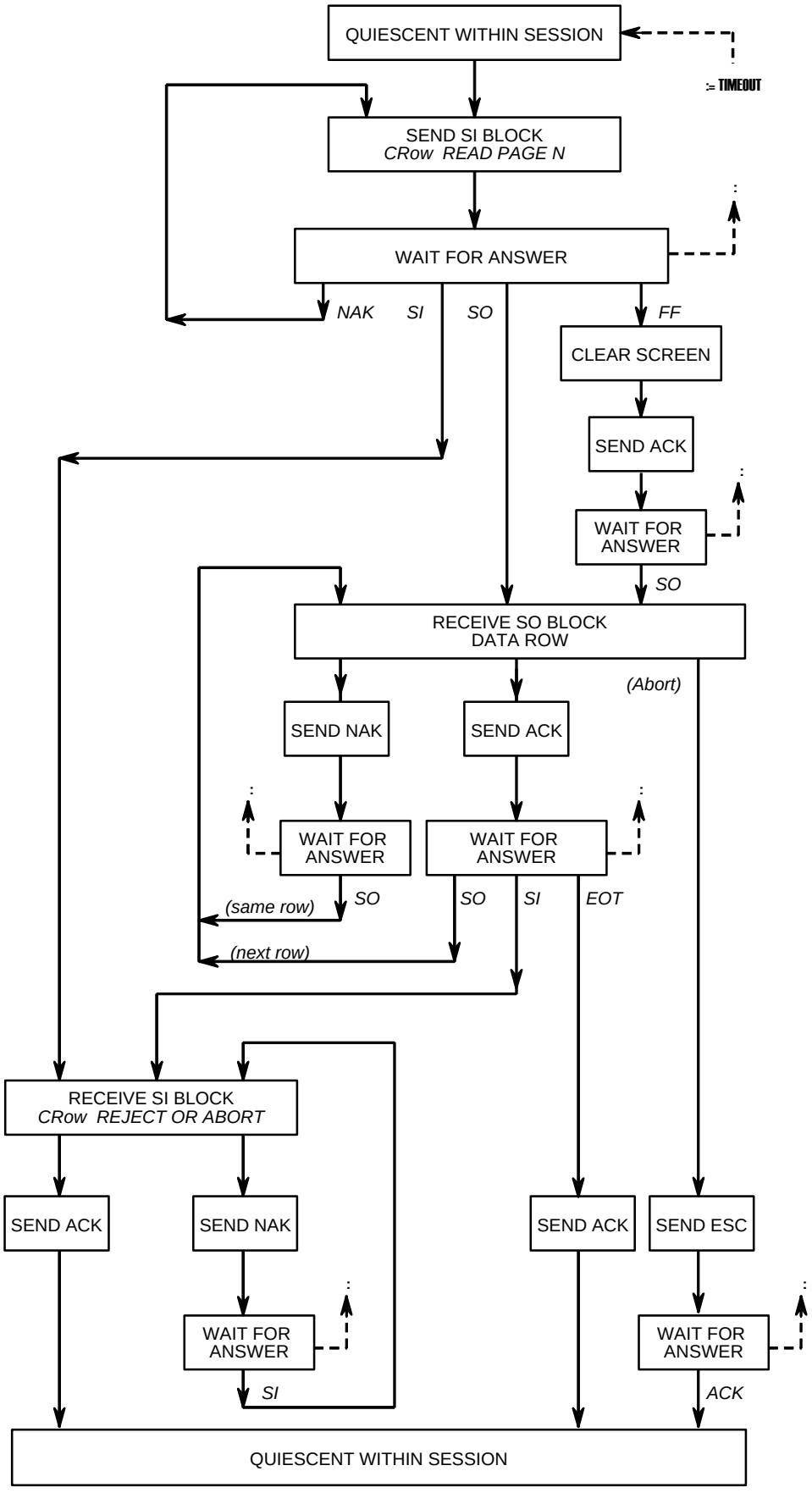


Fig. 5 - Fixed-format page delivery protocol in case of READ PAGE N transfer command

NN		*Reason*
20	-	No reason
21	-	Busy
22	-	Overload
23	-	Erroneous command
24	-	Page unknown
25	-	Unprocessable data

3.6. Fixed-format exchange protocol at session level

Note: This is the former *Section 2.4.*, deferred to this place for reasons of clarity.

a) A LOGIN command in fixed-format is defined by the following SI block:

SI / 24 / I user identifier (, password) ——— / CRC

The user identifier field and the optional password field can contain up to 16 ASCII character codes each.

A positive reaction to such a LOGIN command is given by the transmission of an ACK, whereafter *quiescent state within session* will be assumed.

Negative reactions to such a LOGIN command are given by the transmission of a NAK, which implies retransmission of the LOGIN command, or by:

SI / 24 / NN / ——————————— / CRC

where NN specifies the reason for rejection of the LOGIN command according to the following table:

NN	-	*Reason*
10	-	No reason
11	-	Erroneous command
12	-	User identifier unknown
13	-	Password false

This SI block asks for an ACK or a NAK, whereafter either *quiescent state on the communication line* will be assumed (if ACK) or retransmission is needed (if NAK).

b) A LOGOUT command in fixed-format is defined by the following SI block:

SI / 24 / 0 user identifier ——————— / CRC

The user identifier field can contain up to 16 ASCII character codes.

A positive reaction to such a LOGOUT command is given by the transmission of an ACK, whereafter *quiescent state on the communication line* will be assumed.

Negative reactions to such a LOGOUT command are given by the transmission of a NAK, which implies retransmission of the LOGOUT command, or by:

SI / 24 / NN ——————————— / CRC

where NN specifies the reason for rejection of the LOGOUT command according to the following table:

NN	-	*Reason*
30	-	No reason
31	-	LOGOUT not allowed
32	-	User identifier unknown

This SI block asks for an ACK or a NAK, whereafter either *quiescent state within session* will be assumed (if ACK) or retransmission is needed (if NAK).

3.7. Timeouts

When the master or the slave are awaiting an answer and they do not receive this answer - or they do not receive a complete answer - after a given time called “timeout”, they must return to the quiescent state. The duration of the timeout depends on the implementation; however, a value greater than 10 seconds is proposed.

4. Description of the transmission format

For the transmission of commands and data, a half duplex asynchronous mode is used. The baud rate can be any authorized value between 300 baud and 9600 baud. Every data burst consists of 1 start bit, 8 data bits and 1 stop bit. The bit transmission order is as follows:

Start bit	-	Logical “zero”
D1	-	Least significant bit
D2	-	Data bit
D3	-	Data blt
D4	-	Data bit
D5	-	Data bit
D6	-	Data bit
D7	-	Data bit
D8	-	Most significant bit
Stop bit	-	Logical “one”

5. Transcoding of protocol and teletext format

5.1. Transcoding of protocol

The two protocols have been designed to have the same main functionalities and it is therefore possible to transcode messages from one protocol to another using an intelligent interface, such as a microcomputer, or a dedicated software implemented in the host computer.

5.2. Transcoding of teletext format

It has been recognized that a perfect transcoding between levels one and two of the fixed-format system and variable-format system is not always possible, but that with some default rules, it is possible to obtain a good approximation of the pages when transcoding. The results can be very good when transcoding mosaic graphic pages such as maps.

5.3. Hardware and software of an intelligent terminal

Several implementations can be realized either by modifying the software of existing equipments or by interconnecting those equipments through a microcomputer with a dedicated software. However, to minimize the software production, one solution can be to use a unique microcomputer. In that case, the protocol software can also be used to convert a particular protocol to one of the EBU protocols.

Bibliography

[1] EBU Interim Technical document SPB 492 (1992): **Teletext specification (625 line television systems)**