



INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

V.42 *bis*

**DATA COMMUNICATION
OVER THE TELEPHONE NETWORK**

**DATA COMPRESSION PROCEDURES FOR
DATA CIRCUIT TERMINATING EQUIPMENT
(DCE) USING ERROR CORRECTION
PROCEDURES**

Recommendation V.42 *bis*



Geneva 1990



INTERNATIONAL TELECOMMUNICATION UNION

CCITT

THE INTERNATIONAL
TELEGRAPH AND TELEPHONE
CONSULTATIVE COMMITTEE

V.42 *bis*

**DATA COMMUNICATION
OVER THE TELEPHONE NETWORK**

**DATA COMPRESSION PROCEDURES FOR
DATA CIRCUIT TERMINATING EQUIPMENT
(DCE) USING ERROR CORRECTION
PROCEDURES**

Recommendation V.42 *bis*



Geneva 1990

FOREWORD

The CCITT (the International Telegraph and Telephone Consultative Committee) is the permanent organ of the International Telecommunication Union (ITU). CCITT is responsible for studying technical, operating and tariff questions and issuing recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The Plenary Assembly of CCITT which meets every four years, establishes the topics for study and approves Recommendations prepared by its Study Groups. The approval of Recommendations by the members of CCITT between Plenary Assemblies is covered by the procedure laid down in CCITT Resolution No. 2 (Melbourne, 1988).

Recommendation V.42 *bis* was prepared by Study Group XVII and was approved under the Resolution No. 2 procedure on the 31st of January 1990.

© ITU 1990

All rights reserved. No part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from the ITU.

Recommendation V.42 bis**DATA COMPRESSION PROCEDURES FOR DATA CIRCUIT TERMINATING
EQUIPMENT (DCE) USING ERROR CORRECTING PROCEDURES**

The CCITT,

considering

- (a) that the use of V-Series DCEs for transmission of asynchronous data on the general switched telephone network (GSTN) is widespread;
- (b) that Recommendation V.42 [1] defines error correction procedures providing improved error performance;
- (c) that improved throughput is possible through the use of data compression procedures;
- (d) that there is a need to interwork with DCEs not providing data compression;

declares the view

that the data compression procedures to be followed by DCEs using the error correcting procedures defined in Recommendation V.42 be as specified in this Recommendation.

1 Scope

1.1 General

This Recommendation describes a data compression procedure for use with V-Series DCEs.

The principal characteristics of the data compression procedure are:

- a) a compression procedure based on an algorithm which encodes strings of characters received from data terminal equipment (DTE)
- b) a decoding procedure which recovers the strings of characters from received codewords;
- c) an automatic transparent mode of operation when uncompressible data is detected.

An exploration of the parameters used in this Recommendation is given in § 10.

1.2 Requirements for error correcting procedures

For correct operation of the data compression function it is necessary that an error correcting procedure be implemented between the two entities using this Recommendation. In the case of V-series Recommendations, this requires that the LAPM (link access procedure for modems) error correcting procedures defined in Recommendation V.42 or the error correcting procedures in Recommendation V.120 [2] be implemented.

Note – Undetected bit errors will cause misoperation of the data compression function. Use of a 32-bit frame check sequence (FCS) as defined in ISO 3309 [3] substantially reduces the possibility of such errors. It may therefore be desirable to use this 32-bit FCS (which is an option in V.42 LAPM) in environments with severe impairments.

1.3 A DCE employing data compression

The data compression function may be used with an error-correcting DCE, as shown in Figure 1/V.42 bis. The elements of an error correcting V-series DCE are specified in Recommendation V.42.

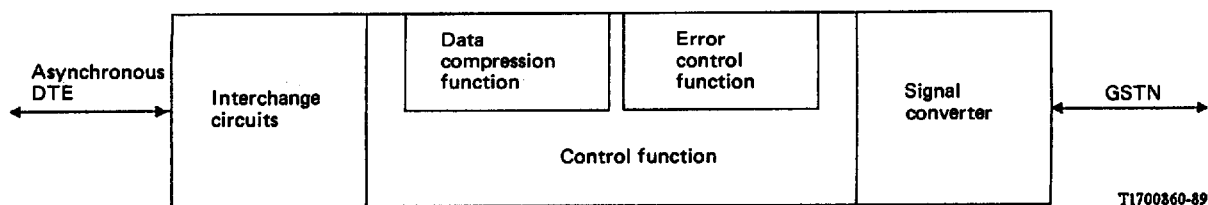


FIGURE 1/V.42 bis

DCE employing data compression and error control**2 Definitions****2.1 character**

Single data element, encoded using a predefined number of bits ($N_3 = 8$).

2.2 start-stop or asynchronous format

Start-stop or asynchronous format is defined in Recommendations V.7 [4] and V.14 [5].

2.3 ordinal value

The ordinal value of a character is the numerical equivalent of the binary encoding of the character. For example, the character "A" when encoded as 01000001 would have an ordinal value of 65_{10} .

2.4 alphabet

Set of all possible characters which may be sent or received across the DTE/DCE interface. It is assumed in this Recommendation that the ordinal values of the alphabet are contiguous from 0 to $N_4 - 1$, where N_4 is the number of characters.

2.5 codeword

A codeword, within the context of this Recommendation, is a binary number in the range 0 to $N_2 - 1$ which represents a string of characters in compressed form. A codeword is encoded using a number of bits, C_2 , where C_2 is initially 9 (i.e. $N_3 + 1$) and increases to a maximum of N_1 bits (see § 7).

2.6 control codeword

A control codeword is reserved for use in DCE-to-DCE signalling of control information related to the compression function whilst in the compressed mode of operation (see § 9).

2.7 command code

Octet which is used for DCE-to-DCE signalling of control information related to the compression function whilst in the transparent mode of operation. Command codes are distinguished from normal characters by a preceding escape character (see § 2.13).

2.8 tree structure

Abstract data structure which is used in this Recommendation to represent a set of strings with the same initial character (see Figure 2/V.42 *bis* and § 6.1).

2.9 leaf node

Point on a tree which represents, within the context of this Recommendation, the last character in a string (see § 6.1).

2.10 root node

A root node is a point on a tree which represents, within the context of this Recommendation, the first character in a string (see Figure 2/V.42 *bis* and § 6.1).

2.11 compressed operation

Compressed operation has two modes. Transitions between the modes may be automatic based on the content of the data received from the DTE (see § 7.1).

2.11.1 compressed mode

A mode of operation in which data from the DTE is transmitted in codewords.

2.11.2 transparent mode

A mode of operation in which compression has been selected but data is being transmitted in uncompressed form. Transparent mode command code sequences may be inserted into the data stream.

2.12 uncompressed operation

A mode of operation in which compression has not been selected. The data compression function is inactive.

2.13 escape character

Within the context of this Recommendation, the escape character is a character which, in transparent mode, indicates the beginning of a command code sequence. This has an initial value of zero, and is adjusted on each appearance of the escape character in the data stream from the DTE, whether in transparent mode or compressed mode (see § 9.2).

3 Abbreviations

The abbreviations introduced in this Recommendation are:

EID Escape in Data, a command code defined in § 9.

ETM Enter Transparent Mode, a control codeword defined in § 9.

ECM Enter Compressed Mode, a command code defined in § 9.

4 Overview of the operation of a DCE incorporating a data compression function

4.1 General

A DCE employing data compression, as depicted in Figure 1/V.42 *bis*, contains the following components:

- a) DTE/DCE interchange circuits;
- b) a signal converter;
- c) a control function;

- d) an error control function; and
- e) a data compression function.

The control function shall have additional capabilities beyond those needed for an error correcting DCE as described in Recommendation V.42. The additional capabilities of the control function are described in § 5, and the operation of the data compression function is described in §§ 6 to 9. The remainder of this section provides an overview of the control function and data compression function.

4.2 *Overview of the control function*

The control function shall perform, in addition to the functions defined in § 6.2 of Recommendation V.42, the following aspects of operation;

- a) negotiation of the presence of the data compression function in the remote DCE, and of parameters associated with the operation of the data compression function;
- b) initialization or re-initialization of the data compression function;
- c) coordination of the establishment of an error controlled connection for use by the peer data compression functions;
- d) coordination of the delivery of data between the DTE/DCE interface and the data compression function, in accordance with the procedures defined in Recommendation V.42, §§ 6.2 and 8.4, including the provision of the flow control procedures defined therein;
- e) coordination of the delivery of data between the data compression function and the error control function;
- f) action on detection of an exception condition.

4.3 *Overview of the data compression function*

The data compression function shall implement the procedures defined in this Recommendation, which result in the efficient encoding of data prior to transmission over the error controlled connection, and shall have the following capabilities:

- a) initialization of the data compression function;
- b) data compression encoding and decoding;
- c) a mechanism for switching between compressed and transparent modes of operation.

4.4 *Communication between the control function and the data compression function*

Communication between the control function and the data compression function is modelled as a set of abstract primitives of the form X-NAME-TYPE which represent the logical exchange of information and control to accomplish a task or service. In the context of this Recommendation the control function is viewed as the "service user" while the data compression function is viewed as the "service provider". The types of primitive are request, indication, response and confirm.

The services expected by the control function are shown in Table 1/V.42 bis.

5 *Operation of the control function*

5.1 *Negotiation of the data compression function*

The use of the data compression function and the associated parameters shall be negotiated at link establishment via a protocol (for example, using the XID procedure defined in Recommendation V.42), following which they remain unchanged for the duration of the error corrected connection.

TABLE 1/V.42 bis
Services expected by the control function

Service	Primitive	§
Initialize the data compression function	C-INIT	5.2, 5.6
Indicate an error to the control function	C-ERROR	5.8
Transfer uncompressed data to/from the data compression function	C-DATA	5.4
Transfer compressed data to/from the data compression function	C-TRANSFER	5.5
Flush remaining untransmitted data from the encoder	C-FLUSH	5.7

Parameter P_0 specifies whether or not compression is to be used. This parameter also specifies the directions (transmit only, receive only, or both directions). The default value of P_0 is 0, indicating no compression in either direction. If compression is proposed for only one direction, then the only valid response is for the proposed direction or no compression. If compression is proposed for both directions, then valid responses are for both directions, for either single direction, or for no compression.

Parameter P_1 represents a proposed value of N_2 the total number of codewords. P_1 shall have a default value of 512, which is its minimum value; a maximum value is not specified within this Recommendation. Any attempt to specify less than the minimum value shall be considered a procedural error and result in disconnection. When values of P_1 are exchanged during the negotiation procedure in one or both directions of transmission, the lower value shall be selected and assigned to N_2 in both DCEs.

Note – See Appendix II for guidance on the choice of value of N_2 , and its effect on performance.

Parameter P_2 is the proposed value for N_7 , the maximum string length. The default value of P_2 is 6, and the permitted range is from 6 to 250. The values outside this range are invalid; and attempt to specify such values shall be regarded as a procedural error and result in disconnection. When values of P_2 are exchanged during the negotiation procedure, the lower value shall be selected and assigned to N_7 in both DCEs.

5.2 Initialization of the data compression function

Following successful negotiation of data compression parameters, the control function shall issue the C-INIT request primitive to the data compression function. The primitive shall indicate the values of the negotiated parameters.

5.3 Connection establishment

On receipt of the C-INIT confirm primitive from the data compression function, the control function shall indicate to the DTE that data transfer may commence.

5.4 Coordination of the transfer of data between the DTE/DCE interface and data compression function

On completion of connection establishment, the control function shall request encoding of the data received on the DTE/DCE interface.

To encode data, the control function shall issue a C-DATA request primitive to the data compression function. This primitive shall indicate the data to be encoded.

On receipt of a C-DATA indication primitive from the data compression function, the control function shall deliver the decoded data to the DTE/DCE interface.

Flow control procedures will be necessary in order to avoid potential loss of data due to buffer overflow. When the procedures defined in this Recommendation are used in conjunction with those defined in Recommendation V.42, the flow control procedures defined in Recommendation V.42, §§ 7.3.1 and 8.4.2, shall be applied.

5.5 *Coordination of the transfer of data between the data compression function and error control function*

On receipt of a C-TRANSFER indication primitive from the data compression function, the control function shall issue an L-DATA request primitive to the error control function.

On receipt of an L-DATA indication primitive from the error control function, the control function shall issue a C-TRANSFER request primitive to the data compression function.

5.6 *Reinitialization of the data compression function*

The control function shall issue a C-INIT request to the data compression function on the following conditions:

- a) L-ESTABLISH indication or confirm;
- b) L-SIGNAL indication or confirm, where the primitive indicates a destructive form.

It is the responsibility of the control functions to ensure that C-INIT request primitives are issued only when no data is in transit between the data compression functions (e.g. in the error control functions) to ensure synchronization between the encoders and decoders.

5.7 *Expedited data transfer*

Certain conditions, the specification of which is outside the scope of this Recommendation, may arise which require that any partially encoded data is transferred immediately, for example if the error control function is in an idle condition. If such a condition arises, the control function shall issue a C-FLUSH request primitive to the data compression function, and shall then transfer remaining data in accordance with § 5.5.

5.8 *Action on detection of C-ERROR*

The C-ERROR indication is used to inform the control function that an error (for example, a procedural error or loss of synchronization) has been detected by the data compression function. The control function shall take appropriate recovery action, including re-establishment of the error corrected connection.

The following conditions recognized by the decoder result in the generation of a C-ERROR indication primitive:

- a) receipt of a STEPUP codeword when it would cause the value of C_2 to exceed N_1 ;
- b) receipt of a codeword, at any time, equal to C_1 ;
- c) receipt of a codeword representing an empty dictionary entry;
- d) receipt of a reserved command code.

6 **Procedures for dictionary use and maintenance**

6.1 *General*

The data compression function employs an algorithm in which a string of characters read from the DTE is encoded as a fixed length codeword. The process employs dictionaries, in which the strings are stored, and which are dynamically updated during normal operation.