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.

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.

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 a 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.

The data compression function contains two dictionaries, one maintained by the data compression encoder for use in compression of data received from the DTE, and one maintained by the data compression decoder for use in the decoding of data received from the error control function.

The dictionary functions are:

a) string matching, in which a sequence of characters is read from the DTE, and the dictionary searched for the resulting string (see § 6.3);

b) updating, in which a new string is added to the dictionary (see § 6.4);

c) the deletion of infrequently used strings in order that storage capacity may be reused (§ 6.5).

The dictionary used to store strings for use in the encoding and decoding process may be logically represented using an abstract data structure. The dictionary can be considered to contain a set of trees, as shown in Figure 2/V.42 *bis*, each with a root corresponding to a character in the alphabet. With the 8-bit character format there will be 256 trees.

A tree represents the set of known strings beginning with one specific character, and each node or point in the tree represents one of this set of strings. The trees in Figure 2/V.42 *bis* represent the strings A, B, BA, BAG, BAR, BAT, BI, BIN, C, D, DE, DO and DOG.

A node that has no dependant nodes, represented by the hierarchically lower level in the tree, is a leaf node. A leaf node represents the last character in a string.

A node that has no parent, represented by the hierarchically higher level in the tree, is a root node. A root node represents the first character in a string.

Associated with each node is a codeword used to uniquely identify the node. The assignment of codewords within the encoder dictionary of a data compression function, and the corresponding assignment of codewords within the decoder dictionary of the peer data compression function in the remote DCE are equivalent, and the codeword thus provides a reversible encoding of a string.

6.2 *Dictionary initialization procedure*

On receipt of a C-INIT request primitive from the control function, the data compression function shall reset the encoder and decoder dictionaries to the initial condition.

In the initial condition, each tree in the dictionary shall consist only of a root node. The codeword associated with each root node shall be N_6 (the number of control codewords) plus the ordinal value of the character represented by the node. The counter C_1, used in the allocation of new nodes (see § 6.5), shall be set to N_5.

6.3 *String matching procedure*

This procedure has the function of matching a sequence of characters (string) with a dictionary entry. The procedure shall commence with a single character representing the first character in the string. The following steps are then applied:

a) a string shall be formed from the first character;

b) if the string matches a dictionary entry, and the entry is not that entry created by the last invocation of the string matching procedure, then the next character shall be read and appended to the string and this step repeated;

c) If the string does not match a dictionary entry or matches the entry created by the last invocation of the string matching procedure, the last character appended to the string shall be removed. The string thus shortened represents the longest matched string and the last character represents the unmatched character.

This procedure will normally match the longest string of characters. However, there are two cases in which step b) shall be terminated before a longest match is found:

i) if an exception condition occurs such as a C-INIT request primitive or C-FLUSH request primitive (while in compressed mode only);

ii) when a transition between transparent and compressed modes of operation occurs.

FIGURE 2/V.42 *bis*

Tree based representation of the dictionary

When in transparent mode the encoder shall use only the criteria specified above for terminating the string matching procedure. When in compressed mode, however, the encoder may employ other criteria for terminating the procedure (e.g. a timeout).

If the string matching procedure is terminated before a longest match is found, the next character from the DTE shall be considered to be the "unmatched character" for the purposes of updating the dictionary and restarting the string matching procedure.

6.4 *Procedure for adding strings to the dictionary*

In order to maintain efficient compression, the dictionary is adapted by the addition of new strings. A new string shall be formed by appending a single character to an existing string, thereby adding a new node onto a tree. The single character shall be the unmatched character resulting from the string matching operation, or the prefix character resulting from the string decoding operation. Following this procedure, the single character required to restart the string matching procedure will be the unmatched character.

There are two conditions under which a new string shall <u>not</u> be added:

a) if this would result in the maximum string length, N_7, being exceeded;

b) if the string is already in the dictionary.

Immediately after the creation of a dictionary entry, the procedure for recovering a dictionary entry shall be applied.

6.5 *Procedure for recovering a dictionary entry*

This section defines a systematic procedure for recovering dictionary entries for re-use when all available entries have been filled. When the last available dictionary entry has been assigned, this procedure recovers a single entry, maintaining the association between the empty entry and its codeword.

A counter C_1 indicates the codeword associated with the next empty dictionary entry, and is maintained in the range N_5 to N_2 - 1. Counter C_1 shall be set to N_5 initially.

The procedure shall be applied only after the creation of a new dictionary entry, and shall consist of the following steps:

a) counter C_1 shall be incremented;

b) if the value of C_1 exceeds N_2 - 1 then C_1 shall be set to N_5;

c) if the node identified by the codeword with value C_1 is in use and not a leaf node, then go to step a);

d) if the node is a leaf node, then it shall be detached from its parent.

7 Operation of the encoding function

7.1 *General*

The encoding function has five principal operations:

a) string matching, in which a sequence of characters from the DTE is matched with a dictionary entry (see § 7.3);

b) encoding, in which the codeword of the matched dictionary entry is represented as a binary value of length C_2 bits (see § 7.4);

c) transfer, in which either the codeword(s) in compressed mode or the characters in transparent mode are passed to the control function (see § 7.5);

d) dictionary updating, in which a new dictionary entry is created, using the matched dictionary entry and the unmatched character (see § 7.6);

e) nod recovery, in which a dictionary entry is recovered for use in the next dictionary update (see § 7.7).

The encoding function operates in one of two modes, transparent mode and compressed mode, switching between these modes on the basis of the test applied in f) below. The sequence of operations, and the cycling of the escape character (see § 9) are identical in the two modes of operation.

The encoder shall support two further operations, which shall be applied only during the string matching procedure in accordance with § 6.3:

f) data compressibility testing, in which the efficiency of the encoding process is estimated and transparent mode or compressed mode selected to maximize efficiency (see § 7.8);

g) flush, in which a C-FLUSH request from the control function indicates that all outstanding data shall be sent (see § 7.9).

7.2 *Initial conditions*

On receipt of a C-INIT request the data compression function shall initialize the encoder to the following state:

a) the dictionary shall be set to the initial condition described in § 6.2;

b) the codeword size C_2 shall be set to $N_3 + 1$;

c) the threshold C_3 shall be set to N_4 X 2;

d) the function shall be set to transparent mode;

e) the escape character shall be assigned the ordinal value 0.

7.3 *String matching*

On receipt of a C-DATA request the data compression function shall apply the string matching procedure defined in § 6.3. The initial character required shall be the unmatched character resulting from the most recent invocation of this procedure.

7.4 *Encoding*

This procedure is used when in the compressed mode of operation. Its purposes is to represent the codeword as a sequence of C_2 bits; the order and numbering of the bits is shown in Figure 3/V.42 *bis*.

If the codeword corresponding to the matched dictionary entry is numerically equal to or greater than the threshold C_3 then:

a) the STEPUP control codeword shall be encoded and transferred using the current codeword size (C_2);

b) the codeword size C_2, shall be increased by 1;

c) multiply C_3 by 2;

d) if the codeword is still numerically greater than or equal to C_3, steps a) to c) shall be repeated.

The codeword is then transferred to the control function, in accordance with the procedures defined in § 7.5.

FIGURE 3/V.42 *bis*

Mapping of codewords into octets

7.5 *Transfer*

In transparent mode, characters shall be passed to the control function for transmission in octet aligned form, using a C-TRANSFER indication. They may be transferred individually during the string matching procedure, or as a sequence following completion of the string matching procedure.

In compressed mode, the matched string shall be encoded according to the procedure defined in § 7.4 and passed to the control function in packed form, with the least significant bit of a codeword immediately following the most significant bit of the preceding codeword.

When the encoder changes state from transparent to compressed mode, the least significant bit of the first codeword to be transferred shall be bit 1 of the next octet position.

Following transfer of a FLUSH control codeword, or when the encoder changes state from compressed to transparent mode following transfer of the ETM control codeword (see § 9) in the sequence, sufficient 0 bits shall be transmitted to ensure that the next transmitted character is octet aligned.

Figure 3/V.42 *bis* provides an example of the data stream passed to the error control function during a transition from compressed to transparent mode. Two 11-bit codewords A and

B are transmitted in compressed form followed by a transition to transparent mode. In this example, the transition requires insertion of seven 0 bits in order that the first uncompressed character, C sent in transparent mode, is octet aligned.

7.6 *Dictionary updating*

A new dictionary node shall be created from the match string and corresponding unmatched character returned by the string matching procedure, using the procedures defined in § 6.4.

7.7 *Node recovery*

Following the creation of a new dictionary node, the node recovery procedure defined in § 6.5 shall be applied.

7.8 *Data compressibility test*

The data compression function shall periodically apply a test to determine the compressibility of the data. The nature of the test is not specified in this Recommendation; however it would consist of a comparison of the number of bits required to represent a segment of the data stream before and after compression.

7.8.1 *Transition to compressed mode*

If the data compression function is in the transparent mode and determines that data compression would be effective, it shall:

a) perform the dictionary update procedure using the current accumulated string and the next character to be processed by the string matching procedure (which will be the first character of the string represented by the first codeword transmitted in compressed mode);

b) indicate to the peer data compression function that a transition to compressed mode is required, using the ECM transparent mode command sequence (see § 9.1);

c) enter compressed mode.

7.8.2 *Transition to transparent mode*

If the data compression function is in the compressed mode and determines that the data stream is currently not compressible, it shall:

a) ensure that the codeword representing any partially encoded data has been transferred in accordance with the procedure given in §§ 7.4 and 7.5;

b) perform the dictionary update procedure using the current accumulated string and the next character to be processed by the string matching procedure (which will be the first character transmitted in transparent mode);

c) indicate to the peer data compression function by transferring the ETM control codeword (see § 9) a transition to transparent mode;

d) transmit sufficient 0 bits to recover octet alignment (see § 7.5);

e) change the state to transparent mode.

7.8.3 *RESET function*

In transparent mode the RESET command code may be used to indicate to the peer data compression function that the encoder dictionary is about to be re-initialized according to the procedures given in §§ 6.2 and 7.2. The RESET command code is sent using the escape character value before re-initialization takes place.

The circumstances under which the encoder requests a dictionary reset are not defined in this Recommendation, but would generally result from the encoder determining that some improvement in performance would result from resetting the dictionary. The procedures for requesting re-initialization of the dictionary on link establishment, or on detection by the control function of an error condition, are defined in §§ 5.2 and 5.6.

The RESET command code is not sent when a C-INIT request primitive is received from the control function.

7.9 *Action on receipt of C-FLUSH request*

Upon receipt of a C-FLUSH request from the control function, if the encoder is in compressed mode and there is a partially-matched string being processed, the data compression function shall:

a) ensure that the codeword representing any partially-matched string is transferred in accordance with the procedures defined in §§ 7.4 and 7.5;

b) perform the dictionary update procedure using the current accumulated string and, when available, the next character to be processed by the string matching procedure;

c) if step a) leaves extra bits pending for transmission (octet alignment not yet achieved), then:

   i) transfer the FLUSH codeword (see § 9);

   ii) if necessary, transfer sufficient 0 bits to recover octet alignment (see § 7.5).

If the encoder is in transparent mode, on receipt of a C-FLUSH request from the control function, the data compression function shall transfer all outstanding data; the string matching procedure is not terminated, and the dictionary update procedure is not performed.

8 Operation of the decoding function

The decoding function shall be capable of operation in both compressed and transparent modes, and shall operate in a manner consistent with that defined in §§ 6, 7 and 9.

On receipt of a C-INIT request from the control function or a RESET command code from the peer data compression function, the data compression function shall initialize the decoding function in accordance with the procedures defined in §§ 6.2 and 7.2.

In transparent mode, the decoding function shall apply the string matching procedure given in § 6.3, in order that the decoder dictionary may be maintained in a compatible state to the peer (remote) encoder dictionary. On receipt of the ECM or EID command code, the decoding function shall operate in a manner consistent with the encoder operations defined in §§ 7.8.1 and 9.2. New dictionary entries shall be created in a manner consistent with the procedures defined in §§ 6.4 and 7.3.

In compressed mode the decoding function shall recover the encoded strings. On receipt of the ETM or FLUSH codewords the decoder shall operate in a manner consistent with the encoder operations defined in §§ 7.8.2 and 7.9. New dictionary entries shall be created using the procedure defined in § 6.4, with the first (prefix) character of the most recently decoded string being appended to the previous decoded string.

The decoder shall regard the STEPUP control codeword as an indication that the encoder has increased the codeword size in accordance with the procedures defined in § 7.4.

9 Communications between peer data compression functions

9.1 *Control codewords and command codes*

The control codewords and command codes allocated for communication between peer data compression functions are given in Table 2/V.42 *bis*.

9.2 *Procedures for use of the escape sequence*

A transparent mode command sequence shall consist of the escape character followed by one of the command codes listed in § 9.1 above.

To reduce data expansion resulting from the escape mechanism defined below, if the current escape character is detected within the data stream from the DTE, the data compression function shall:

a) if in transparent mode, transfer the detected escape character, and transmit the EID code, and then

b) in both transparent and compressed modes, modify the value of the escape character by adding to it the decimal value 51, the addition to be performed modulo 256.

TABLE 2/V.42 *bis*

Control code words (used in compressed mode)		
Code word	Name	Description
0	ETM	Enter transparent mode
1	FLUSH	Flush data
2	STEPUP	Step up codeword size
Command codes (used in transparent mode)		
Value	Name	Description
0	ECM	Enter compression mode
1	EID	Escape character in data
2	RESET	Force reinitialization
3 to 255	Reserved	

10 Parameters

The following parameters are required by the data compression function. N_1 to N_7 and P_0 to P_2 apply to both directions of transmission, whilst a separate set of C_1, C_2, C_3 variables must be provided in the encoder and decoder.

N_1 Maximum codeword size (bits);

N_2 Total number of codewords;

N_3 Character size (bits):

$N_3 = 8$;

N_4 Number of characters in the alphabet:

;

N_5 index number of first dictionary entry used to store a string:

$N_5 = N_4 + N_6$;

N_6 Number of control codewords:

$N_6 = 3$;

N_7 Maximum string length;

C_1 Next empty dictionary entry;

C_2 Current codeword size;

C_3 Threshold for codeword size change;

P_0 V.42 *bis* data compression request;

P_1 Number of codewords (negotiation parameter);

P_2 Maximum string size (negotiation parameter).

ANNEX A

(to Recommendation V.42 *bis*)

PROCEDURES FOR NEGOTIATING V.42 *BIS* WHEN USED WITH V.42

When using this data compression Recommendation with V.42 error control, the XID negotiation procedure shall be used (see §§ 7.6, 8.10, 10 of Rec. V.42 and ISO 8885 - 1987 [6, 7].) A data link layer subfield in addition to those defined in Recommendation V.42 shall be used for this purpose. It shall appear in the XID frame immediately before the user data subfield and shall be encoded as in Table A-1/V.42 *bis*.

During the protocol establishment phase, the presence of parameter P_0 in the Private Parameter Set Data Link Layer Subfield of the XID frame shall indicate a request for data compression.

Note – The incorporation of the contents of this Annex into Recommendation V.42 is under consideration by Study Group XVII.

TABLE 1/V.42 *bis*

	Bit 8 . . . 1	
Group ID	11110000	Private parameter ser (ISO 8885, Addendum 3)
Group Length	nnnnnnnn	(MSB) Length of parameter field (excludes group ID and length)
	nnnnnnnn	(LSB)
Parameter ID	00000000	Parameter set identifier
Parameter length	00000011	Length of string
Parameter value	01010110 00110100 00110010	V 4 2
Parameter ID	00000001	Rec. V.42 *bis* - Data Compression request (P_0)
Parameter length Parameter value	00000001 000000nn	Length of field Request for compression in: 00 neither direction 01 negotiation initiator-responder direction only 10 negotiation responder-initiator direction only 11 both directions
Parameter ID	00000010	Rec.V.42 *bis* - Number of codewords (P_1)
Parameter length	00000010	16-bit integer
Parameter value	nnnnnnnn nnnnnnnn	(MSB) Value of parameter P_1 (LSB)
Parameter ID	00000011	Rec. V.42 *bis* - Maximum string length (P_2)
Parameter length	00000001	8-bit integer
Parameter value	nnnnnnnn	Value of parameter P_2

MSB: Most significant bit
LSB: Least significant bit

APPENDIX I

(to Recommendation V.42 *bis*)

SDL DESCRIPTION OF ENCODER (Z.100 TO Z.104) [8]

Figure I-1/V.42 *bis* shows the set of SDL symbols used in the diagrams of this Appendix.

The following diagrams provide an illustration of the operation of the encoder:

a) Figure I-2/V.42 *bis*: Encoder (see §§ 7.2, 7.3, 7.8, 7.9). This diagram illustrates the operation of the main elements of the encoder.

b) Figure I-3/V.42 *bis*: Process of character (see §§ 6.3, 6.4, 6.5). This diagram illustrates the operation of the string matching procedure, the conditions under which it is terminated and the action taken.

c) Figure I-4/V.42 *bis*: Check codeword size (see § 7.4). This diagram illustrates the codeword size step up mechanism.

d) Figure I-5/V.42 *bis*: Test compression (see § 7.8). This diagram illustrates the procedures for changing between transparent and compressed modes of operation and for use of RESET.

e) Figure I-6/V.42 *bis*: Flush (see § 7.9). This diagram illustrates the action taken on receipt of a C-FLUSH request.

f) Figure I-7/V.42 *bis*: Exception process next character (see §§ 7.8.1 a, 7.8.2 b, 7.9 b). This diagram illustrates the means by which the compressed mode character processing is performed following a transition to compressed mode or a flush operation.

g) Figure I-8/V.42 *bis*: Escape character procedure (see § 9.2).

h) Figure I-9/V.42 *bis*: Signal reset procedure (§ 7.8.3).

i) Figure I-10/V.42 *bis*: Add string + character to dictionary procedure (§§ 6.4 and 6.5).

FIGURE I-1/V.42 *bis*

SDL Symbols used

FIGURE I-2/V.42 *bis*

Encoder

FIGURE I-3/V.42 *bis*

Process character procedure

FIGURE I-4/V.42 *bis*

Check codeword size procedure

FIGURE I-5/V.45 *bis*

Test compression procedure

FIGURE I-6/V.42 *bis*

Flush procedure

FIGURE I-7/V.42 *bis*

Exception process next character procedure

FIGURE I-8/V.42 *bis*

Escape character procedure

FIGURE I-9/V.42 *bis*

Signal "Reset" procedure

FIGURE I-10/V.42 *bis*

Add "string + character" to dictionary procedure

APPENDIX II

(to Recommendation V.42 *bis*)

GUIDANCE FOR IMPLEMENTORS

The following notes provide information on the implementation of the data compression scheme and on the selection of parameters.

II.1 *Selection of N_2, the total number of codewords*

The dictionary size is equal to N_2 - N_6 (assuming that entries are not provided for the reserved codewords). The selection of a large value for N_2 means that the number of strings available is large, but also that the value of N_1 is larger. The gain in performance obtained from the selection of a larger dictionary may be offset by the larger codeword size needed, and for certain types of data, better performance may be obtained by using a smaller dictionary. If values of N_2 in the range $2^n + 1$ (for integer n) to approximately 1.3×2^n are selected, no performance improvement will be gained over the selection of the value 2^n. A value for N_2 of 2048 provides good compression performance across a wide range of data types.

II.2 *Data structures*

The data compression scheme described in this Recommendation is well suited to implementation using a tree based data structure. This type of data structure will provide good utilization of memory space and fast searching.

II.3 *Calculation of compression performance*

The calculation of compression performance may be expressed as the number of characters received by an encoder divided by the number of octets transferred from the encoder (to the error control function). The count of characters and octets should be set to zero on receipt of a C-INIT request.

II.4 *Examples of the operation of the encoder*

The following three examples illustrate the operation of the encoder. It is assumed that the dictionary is in the state shown in Figure 2.V.42 *bis*.

II.4.1 *Simple case: "BAY", compressed mode*

The first character "B" is read, and the dictionary searched for the string "B". As this string is present, the next character "A" is read and appended, forming the new string "BA". The dictionary is searched for the new string and when it is found, the next character "Y" is read and appended, forming the new string "BAY". The dictionary is searched for "BAY", which is not present. "Y" is removed, and the string matching procedure exits with "BA" as the matched string and "Y" as the unmatched character.

The codeword for "BA" is encoded in C_2 bits, packed into octets, and passed to the control function for transmission. The new string "BAY" is created by appending "Y" to "BA", assigning the codeword with value C_1 to this new string. C_1 is incremented, and the node (string) currently assigned this value is tested to see if it is empty or is a leaf node. If the node is empty, it will be used in the next dictionary update. If the node is used and is not a leaf node, i.e. is part of a longer string, then C_1 is incremented again and the test repeated. If the node is a leaf node, then it is detached from its parent and will be re-used in the next dictionary update.

The character "Y" will be used to restart the string match.

II.4.2 *Simple case: "BAY", transparent mode*

In transparent mode, the same sequence of operation described in § II.4.1 will occur, the only difference being that the characters "A", "Y" will be transmitted in place of the codeword for "BA".

II.4.3 *Repeated characters or sequences: "CCCCC", compressed mode*

The aim of this example is to illustrate a particular aspect of the algorithm. As the encoder is able to update its dictionary on the basis of look-ahead, whilst the decoder is only able to update its dictionary on the basis of previously decoded data, it is necessary to ensure that the encoder does not use new dictionary entries before they are transmitted to the decoder.

The first "C" is read and will be matched with the dictionary entry for "C". The second "C" is read, appended to the first, and the dictionary searched for "CC". As "CC" is not in the dictionary, the string matching procedure exits with the matched string as "C" and the unmatched character as "C". "CC" is added to the dictionary, the codeword for "C" sent, and string matching restarts with the second "C".

The third "C" is read, appended to the second "C", forming "CC", and the dictionary searched for "CC". As this is in the dictionary but is, however, the entry created since the last string match, [see § 6.3 b)], the string matching procedure exits with the matched string as "C" and the unmatched character as "C". "CC" is not added to the dictionary as it is already present, the codeword for "C" sent, and string matching starts with the third "C".

The fourth "C" is read, appended to the third "C", forming "CC", and the dictionary searched for "CC" As "CC" is found in the dictionary, and it does not match the entry created since the last string match (the update operation was inhibited), the fifth "C" is read and appended to the string.

References

[1] CCITT Recommendation Error Correcting Procedures for DCEs using Asynchronous-to-Synchronous Conversion, Vol. VIII, Rec. V.42.

[2] CCITT Recommendation *Support by a ISDN of Data Terminal Equipment with V- Series Type Interfaces with Provision for Statistical Multiplexing*, Vol. VIII, Rec. V.120.

[3] ISO 3309 - *Data Communication - High Level Data Link Control Procedures - Frame Structure*.

[4] CCITT Recommendation *Definitions of terms concerning data communication over the telephone network*, Vol. VIII, Rec. V.7.

[5] CCITT Recommendation *Transmission of Start Stop Characters over Synchronous Bearer Channels*, Vol. VIII, Rec. V.14.

[6] ISO 8885 - 1987 *Information Processing Systems - Data Communications - High Level Data Link Control Procedures - General Purpose XID Frame Information Field Content and Format*.

[7] ISO 8885 - 1987/ADD3 *Information Processing Systems - Data Communications - High Level Data Link Control Procedures - General Purpose XID Frame Information Field Content and Format - Addendum 3: Definition of a private parameter data link layer subfield*.

[8] Serie Z.100 CCITT Recommendations *Functional Specification and Description Language (SDL)*, Vol. X.