INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

T.105

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(11/94)

TERMINALS FOR TELEMATIC SERVICES

SYNTAX-BASED VIDEOTEX APPLICATION LAYER PROTOCOL

ITU-T Recommendation T.105

(Previously "CCITT Recommendation")

FOREWORD

The ITU-T (Telecommunication Standardization Sector) is a permanent organ of the International Telecommunication Union. The ITU-T 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 World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1 (Helsinki, March 1-12, 1993).

ITU-T Recommendation T.105 was prepared by the ITU-T Study Group 8 (1993-1996) and was approved under the WTSC Resolution No. 1 procedure on the 11th of November 1994.

NOTE

In this Recommendation, the expression “Administration” is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

CONTENTS

Page

SUMMARY

This Recommendation specifies both:

- the end-to-end Application Layer protocol for the Syntax-based Videotex service; and
- the underlying Bearer Independent Service.

This Recommendation is applicable to all devices supporting the Bearer Independent Service on the upper side of the Network Layer and is therefore network independent.

New or enhanced additional features can be offered compared with the traditional Videotex Services, e.g.:

- direct selection of Videotex applications;
- multi-communication to several hosts or applications;
- multimedia communication (text, geometric, photographic, audio);
- distributed Videotex applications;
- use of supplementary services;
- handling of calls in both directions (outgoing, incoming).

Recommendation T.105

SYNTAX-BASED VIDEOTEX APPLICATION LAYER PROTOCOL

(Geneva, 1994)

1 Scope

This Recommendation specifies both the end-to-end Application Layer protocol for the Syntax-based Videotex Service and the underlying Bearer Independent Service (BIS).

NOTE 1 – This Recommendation does not specify any service aspects of a Syntax-based Videotex Service.

This Recommendation is applicable to all devices supporting the Bearer Independent Service on the upper side of the Network Layer (see clause 11). In the context of this Recommendation, a device is either a Videotex Terminal, a Videotex Service Centre, a Videotex Access Point or a Videotex Host (see 3.1).

NOTE 2 – Depending on the communication type in use (e.g. ISDN DTE/DTE, ISDN DTE/DCE or PSTN), a separate Recommendation is assumed to be applicable for the respective lower layer protocols.

2 Normative references

The following Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision: all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of currently valid ITU-T Recommendations is regularly published.

[1] CCITT Recommendation F.300 (1988), *Videotex Service*.

[2] CCITT Recommendation T.51 (1988), *Coded character sets for telematic services*.

[3] CCITT Recommendation X.3, *Packet assembly/disassembly facility (PAD) in a public data network*.

[4] CCITT Recommendation X.25 (1980), *Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit*.

[5] CCITT Recommendation X.25 (1984 or later), *Interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit*.

[6] CCITT Recommendation X.29, *Procedures for the exchange of control information and user data between a packet assembly/disassembly (PAD) facility and a packet mode DTE or another PAD*.

[7] CCITT Recommendation X.121 (1992), *International numbering plan for public data networks*.

[8] CCITT Recommendation T.101, *International interworking for Videotex services.*

[9] CCITT Recommendation T.102, *Syntax-based Videotex End-to-End protocols for circuit mode ISDN*.

[10] CCITT Recommendation T.103, *Syntax-based Videotex End-to-End protocols for packet mode ISDN*.

[11] CCITT Recommendation T.104, *Packet mode access for syntax-based Videotex via PSTN*.

[12] CCITT Recommendation T.106, *Framework of Videotex terminal protocols*.

[13] CCITT Recommendation X.200, *Reference model of Open Systems Interconnection for CCITT applications*.

[14] ISO 7498, *Information processing systems – Open Systems Interconnection – Basic Reference Model*.

[15] ISO/IEC 8208 (1990), *Information technology – Data Communications – X.25 Packet Layer Protocol for Data Terminal Equipment*.

[16] ISO/IEC 8208:1990/Am.3 (1991), *Information technology – Data Communications – X.25 Packet Layer Protocol for Data Terminal Equipment – Amendment 3: Conformance Requirements*.

[17] ISO/TR 8509, *Information processing systems – Open Systems Interconnection – Service conventions*.

3 Definitions and abbreviations

3.1 Definitions

For the purposes of this Recommendation, the following definitions apply:

access function: The functional entity which gives access to the Videotex Service. This entity is an integral part of the Videotex Service.

access network: The network which provides the link between the Terminal Function and the Access Function.

application layer: See OSI Reference Model CCITT Rec. X.200 [13] **|** ISO 7498 [14].

audio data: Generic term for data which can be made audible (e.g. wave form encoded, phonemically encoded).

bearer independent service access point: Point in an end system where the user of the Bearer Independent Service accesses the service.

called BIS user: A BIS user with whom a calling BIS user wishes to establish a NC.

calling BIS user: A BIS user that initiates a NC establishment request.

confirm: See Service Conventions Standard ISO/TR 8509 [17].

data circuit-terminating equipment: See ISO/IEC 8208 [15] and CCITT Recommendation X.25 [5].

data terminal equipment: See ISO/IEC 8208 [15] and CCITT Recommendation X.25 [5].

distributed videotex application: A Videotex Application which makes use of more than one Videotex Host.

host access network: The network which provides the link between the Access Function and the Host Function(s). It is an integral part of the Videotex Service and may be void.

host function: The abstraction of the Videotex Applications available in a particular Videotex Service.

in-band: All communication aspects related to CCITT Recommendation X.29 [6] (including Telematic Commands) based procedures.

indication: See Service Conventions Standard ISO/TR 8509 [17].

logical channel: See ISO/IEC 8208 [15] and CCITT Recommendation X.25 [5].

multimedia communication: Term used to indicate that both pictorial and audio data are exchanged.

network connection: See OSI Reference Model CCITT Recommendation X.200 [13] | ISO 7498 [14].

network layer: See OSI Reference Model CCITT Recommendation X.200 [13] | ISO 7498 [14].

network service: See OSI Reference Model CCITT Recommendation X.200 [13] | ISO 7498 [14].

out-band: All communication aspects related to procedures not being based on CCITT Recommendation X.29 [6] (including Telematic Commands), e.g. in case of an ISDN the use of the D-channel for B-channel establishment.

packet layer: See ISO/IEC 8208 [15] and CCITT Recommendation X.25 [5].

pictorial data: Generic term for data which can be displayed (e.g. alphamosaic, geometric, photographic).

primitive: See Service Conventions Standard ISO/TR 8509 [17].

request: See Service Conventions Standard ISO/TR 8509 [17].

response: See Service Conventions Standard ISO/TR 8509 [17].

telematic command: In the scope of this Recommendation, a Telematic Command is a specific service element which is carried in a complete packet sequence of X.25 Packet Layer Protocol DATA packets with the Q-bit set to 1. The first octet in the User Data field of the first packet of the complete packet sequence carries a value which is reserved for videotex [see CCITT Recommendation X.29 (1992), “Telematic service message, videotex”].

terminal function: The abstraction of a functional entity which acts as a Videotex Terminal.

videotex access point: See CCITT Recommendation F.300 [1].

videotex application: See CCITT Recommendation F.300 [1].

videotex external host: See CCITT Recommendation F.300 [1].

videotex host: This term describes a computer which offers one or more applications and/or facilities. It can be represented through a Videotex Host Computer, an External Videotex Host or a Videotex Service Centre.

videotex host computer: See CCITT Recommendation F.300 [1].

videotex service: See CCITT Recommendation F.300 [1].

videotex service centre: See CCITT Recommendation F.300 [1].

NOTE – According to CCITT Recommendation F.300, a Videotex Service Centre provides host and/or access functions, i.e. it may also act as a Videotex Access Point.

videotex terminal: See CCITT Recommendation F.300 [1].

virtual call: See ISO/IEC 8208 [15] and CCITT Recommendation X.25 [5].

virtual circuit: See ISO/IEC 8208 [15] and CCITT Recommendation X.25 [5].

3.2 Abbreviations

For the purposes of this Recommendation, the following abbreviations apply:

AP	Access Point
AU	Access Unit
BIS	Bearer Independent Service
CCITT	International Telegraph and Telephone Consultative Committee
CD	Call Deflection
CLIP	Calling Line Identification Presentation
CLIR	Calling Line Identification Restriction
DCE	Data Circuit-terminating Equipment
DDI	Direct-Dialling-In
DDU	Dialogue DU

DFK	Definable Function Key
DS	Data Syntax
DTE	Data Terminal Equipment
DU	Data Unit
DXE	Either a DTE or a DCE
HAN	Host Access Network
IB	In-Band
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
ITU-T	International Telecommunication Union – Telecommunication Standardization Sector (formerly known as “CCITT”)
IVAP	ISDN syntax-based VAP
IVT	ISDN syntax-based VT
LC	Logical Channel
LSB	Least Significant Bit
M-bit	More Data bit
MBS	M-bit Sequence
MSB	Most Significant Bit
MSN	Multiple Subscriber Number
NC	Network Connection
NL	Network Layer
OB	Out-Band
OSI	Open Systems Interconnection
PDN	Public Data Network
PDU	Protocol DU
PH	Packet Handler
PLP	Packet Level Protocol
PSPDN	Packet Switched Public Data Network
PSTN	Public Switched Telephone Network
Q-bit	Qualifier bit
SBV	Syntax-Based Videotex
SUB	Subaddressing
TC	Telematic Command
TCH	TC Header
TCM	TC Message
TDU	Telesoftware DU
TFI	Terminal Facility Identifier
TPD	Transparent Processable Data
UUI	User-to-User Information
UUS	User-to-User Signalling
VAP	Videotex AP
VC	Virtual Call
VH	Videotex Host
VPDE	Videotex Presentation Data Element
VS	Videotex Service
VT	Videotex Terminal

4 Overview

The main feature of a Syntax-based Videotex Terminal (VT) is the capability to access those Videotex Services which are defined for and used in a Public Switched Telephone Network (PSTN) or Packet Switched Public Data Network (PSPDN) environment using the X.25 Packet Layer Protocol (X.25/PLP) on an arbitrary access network.

More advanced Videotex Services may make use of additional features which are presented in the following subclauses.

4.1 Direct selection of Videotex Applications

The protocol allows the Videotex Terminal to shortcut the dialogue and welcome phase of the Videotex Service and to be connected with a given Videotex Application identified by a network address or a mnemonic.

4.2 Multicommunication

Multicommunication allows a terminal to establish several independent communication channels to one or several independent Videotex Hosts or Videotex Applications.

The protocol supports multicommunication using several Network connections.

NOTES

1 In the case of an ISDN, several Network connections may be provided using one B-channel, only.

2 The handling of several independent communication channels may appear in any configuration [see Annex A (informative)].

EXAMPLE: Assume that "ETSI" and "ITU-T" are two Videotex Applications offering the possibility to Videotex users to read texts of telecommunication Recommendations. In order to compare the text of a given Recommendation with an equivalent standard, the terminal should establish a network connection with each of the two applications. The information from the two applications shall be simultaneously displayed on the terminal using some windowing capability.

4.3 Multimedia communication

Multimedia information may contain pictorial data (i.e. alphamosaic, geometric, and photographic display elements) and audio data. The audio data may be transmitted either in parallel or in serial with the pictorial data.

When the audio data is transmitted in serial with the pictorial data, a single Network connection may be used to carry both.

When the audio data has to be sent in parallel with the pictorial data, a second Network connection dedicated to the transmission of audio data shall be setup.

The protocol supports multimedia communication using several Network connections, providing for the simultaneous reproduction of both pictorial and audio information, in addition to a serial transmission already covered by the data syntax.

NOTES

1 In the case of an ISDN, several Network connections may be provided using one B-channel, only.

2 The handling of several additional communication channels may appear in any configuration [see Annex A (informative)].

4.4 Distributed Videotex Application

It is recognized that, to accomplish some specific application, the application may be decomposed into several sub-applications which may be implemented on more than one host. A communication channel from the terminal to each of the sub-applications may exist. The incoming datastreams are combined in the terminal to deliver the desired service to the user.

The protocol supports distributed Videotex Applications using several Network connections.

NOTE – In the case of an ISDN, several Network connections may be provided using one B-channel, only.

EXAMPLE: Assume an application which serves PSTN terminals and ISDN terminals at the same time. The PSTN application uses only alphamosaic information, while the ISDN application additionally offers photographic pictures. The "alphamosaic-host" is used for the dialogue with both PSTN and ISDN terminals. When a specific picture has to be displayed, on request of the "dialogue-host" a second VC is established between the ISDN terminal and the "picture-host".

4.5 Supplementary services

In the case of an ISDN, the protocol allows for a use of several ISDN supplementary services [Multiple Subscriber Number (MSN), Subaddressing (SUB), User-to-User Signalling 1 (UUS 1), Call Deflection (CD), Calling Line Identification Presentation (CLIP) and Calling Line Identification Restriction (CLIR)], mainly to support ISDN-specific Videotex Service selection mechanisms. For details on these supplementary services, refer to Annex B (informative).

NOTE – The use of supplementary services on networks other than an ISDN is for further study.

4.6 Handling of incoming calls

The protocol permits a Syntax-based Videotex Terminal to be called by another terminal or a Videotex Service.

In the case of a called Syntax-based Videotex Terminal, the handling of supplementary services is not within the scope of this Recommendation.

5 Configurations

Various configurations and topologies may be used, examples of which are given in Annex A (informative). It shall be the responsibility of the Videotex Service providers to opt for the appropriate configuration(s) in the definition of the Syntax-based Videotex Service.

6 General model

In order to describe all configurations, the following reference models are used:

- a communication model;
- a terminal model.

The communication model describes the relationships between the functional entities involved in the communication. It models the communication between a user and a Videotex Service in terms of terminal functions, access network functions and service functions.

The terminal model defines, in an abstract way, the terminal aspects which are relevant for the protocol.

6.1 Communication model

In describing the communication model, the following entities can be identified:

- a User;
- a Terminal Function;
- an Access Network;
- an Access Function;
- a Host Access Network;
- a Host Function.

In all Videotex configurations, the **User** employs a terminal to communicate with a Videotex Service. The abstraction of the terminal is named **Terminal Function**.

The **Access Network** provides the link between the Terminal Function and the Videotex Service. It may consist of one or several networks of different types. The Bearer Independent Service (BIS) provides the interface to the access network (see clause 11).

NOTE – The definition of the relation between the BIS and a specific access network is not in the scope of this Recommendation.

The **Access Function** is the functional entity which gives access to the Videotex systems. It is an integral part of the Videotex Service.

The **Host Access Network**, which is an optional functional entity, is the abstraction of the function, which connects the Access Function to one or more Host Functions. Depending on the actual topology used in a specific service, it may be void.

The **Host Function** is the abstraction of the collection of Videotex Applications.

The interfaces and protocols defined between the functional entities are:

(1) User interface

(2) Terminal-Access Network interface

(3) Access Function-Access Network interface

(4) Access Function-Host Access Network interface

(5) Host Function-Host Access Network interface

(6) Terminal-Access Function protocol

(7) Host Access protocol

Figure 1 shows the relationships between the entities.

Of these interfaces and protocols, only protocol (6) is within the scope of this Recommendation, interfaces (2) and (3) are provided by the BIS.

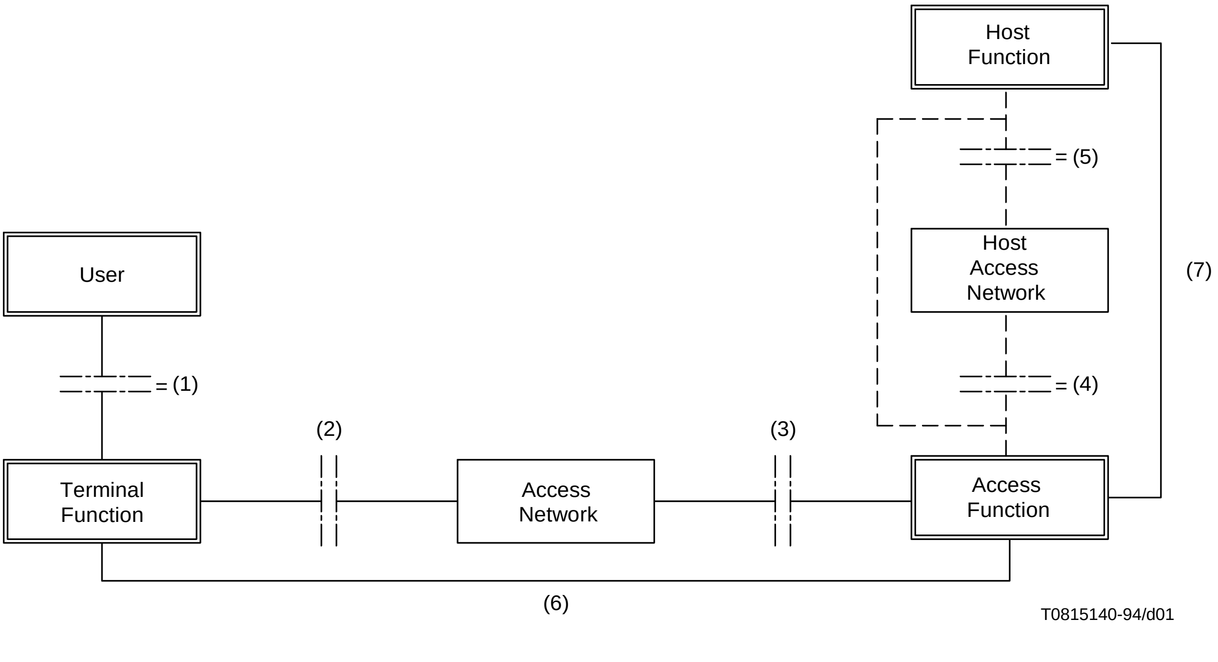


FIGURE 1/T.105

Communication Model

6.2 Terminal model

The terminal model is the reference model which is used throughout the document. This model describes the relationships between functions of a Syntax-based Videotex Terminal.

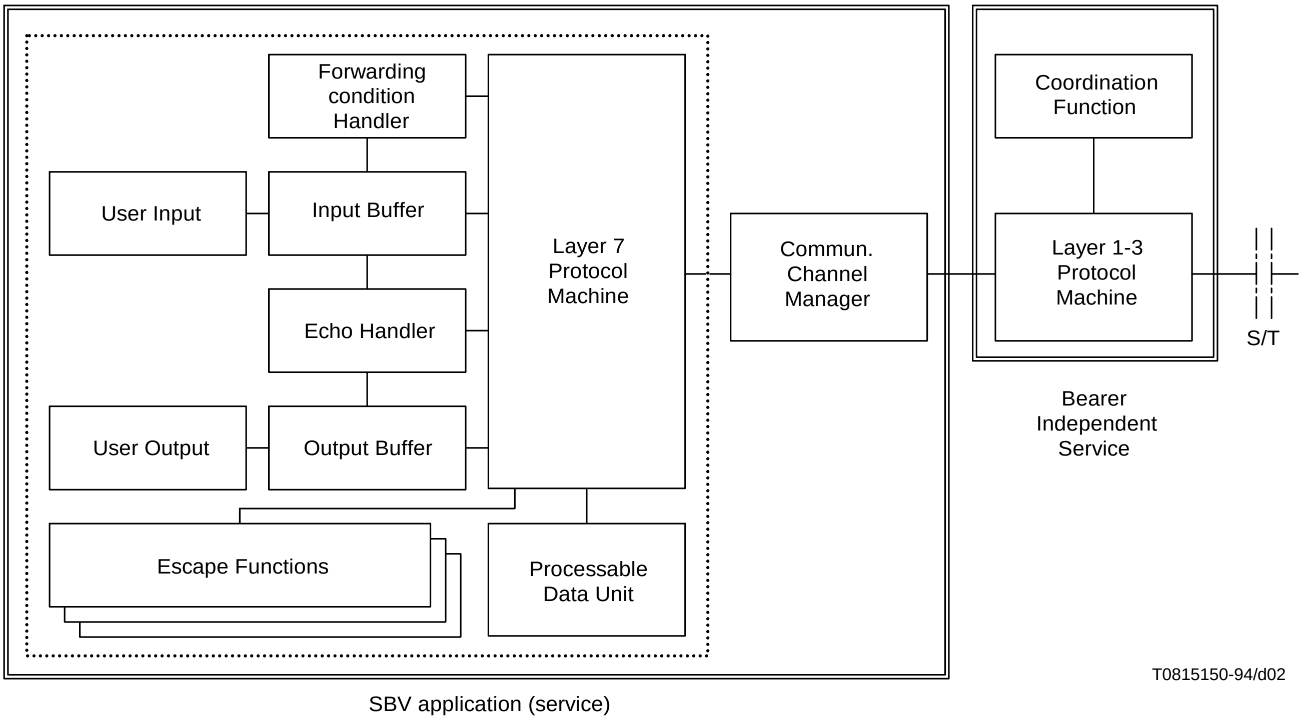
The Application Layer is decomposed into the following components:

- communication channel manager;
- Layer 7 protocol machine;
- input buffer;
- output buffer;
- echo handler;
- forwarding condition handler;
- processable data unit;
- escape function units.

The protocol machine handling Layers 1-3 of the protocol and the co-ordination function are integral parts of the BIS entity and do not require further refinement in the terminal model.

NOTE 1 – Depending on the access network and the communication mode, the co-ordination function may also be empty.

Figure 2 introduces these functions and their relationship.



NOTE – For every communication channel, one separate independent instance of each of the functional entities inside the dotted rectangle exists.

FIGURE 2/T.105

Functional entities of the terminal model

In the context of this Recommendation, a **communication channel** is equivalent to an established Network connection.

The **communication channel manager** supervises the handling of related communication channels. The protocol allows for the definition of a hierarchical relationship between communication channels. An application can request the establishment of further dependent communication channels via an already established communication channel. The requesting communication channel shall be named a "Master"-channel and the requested communication channels shall be named "Slave"-channels.

NOTE 2 – The handling of more than one communication channel is optional. Therefore, the communication channel manager may also be empty.

This relationship is maintained during the lifetime of a communication channel implying that existing "Slave"-channels have to be closed implicitly when the "Master"-channel is closed. The communication channel manager takes care of this implicit closure and keeps track of related communication channels.

NOTE 3 – A communication channel may be a "Slave"-channel and a "Master"-channel at the same moment, provided:

1) it is related to a "Master"-channel;

2) there is at least one "Slave"-channel related to this channel.

For each established communication channel, one separate independent (virtual) instance of the following functional entities exists (regardless of whether it is an independent or a linked communication channel).

The **Layer 7 protocol machine** supports the procedures as defined in the Application Layer protocol (see clause 8). It conveys information to the output buffer (display data), the echo handler (parameter settings, related to echoing), the forwarding condition handler (parameter settings, related to forwarding and function key information), the communication channel manager (related communication channels) or the processable data or escape modules.

The **input buffer** receives the user input, which subsequently is forwarded to the protocol machine in order to be transmitted to the Access Function, and which may be used in parallel by the echo handler. A terminal may locally edit this buffer. The origin of data entered in the input buffer need not be defined, having no relevance for the description of the protocol of this Recommendation.

NOTE 4 – The specification of local editing is outside the scope of this Recommendation.

The **output buffer** receives display data from both the Layer 7 protocol machine and the echo handler. The processing of display data is not described in the model having no relevance for the description of the protocol of this Recommendation.

The **echo handler** controls the echoplex procedures. With regard to echoing, the state diagram in Figure 3 shall apply.

When echoing is enabled, the initial terminal state shall be ECHO_ACTIVE. The initial state of echoing is defined in Annex C (normative).

In the state ECHO_ACTIVE, all user input shall be echoed immediately. Echoing shall be performed on a character basis in the sense of Recommendation T.101 [8].

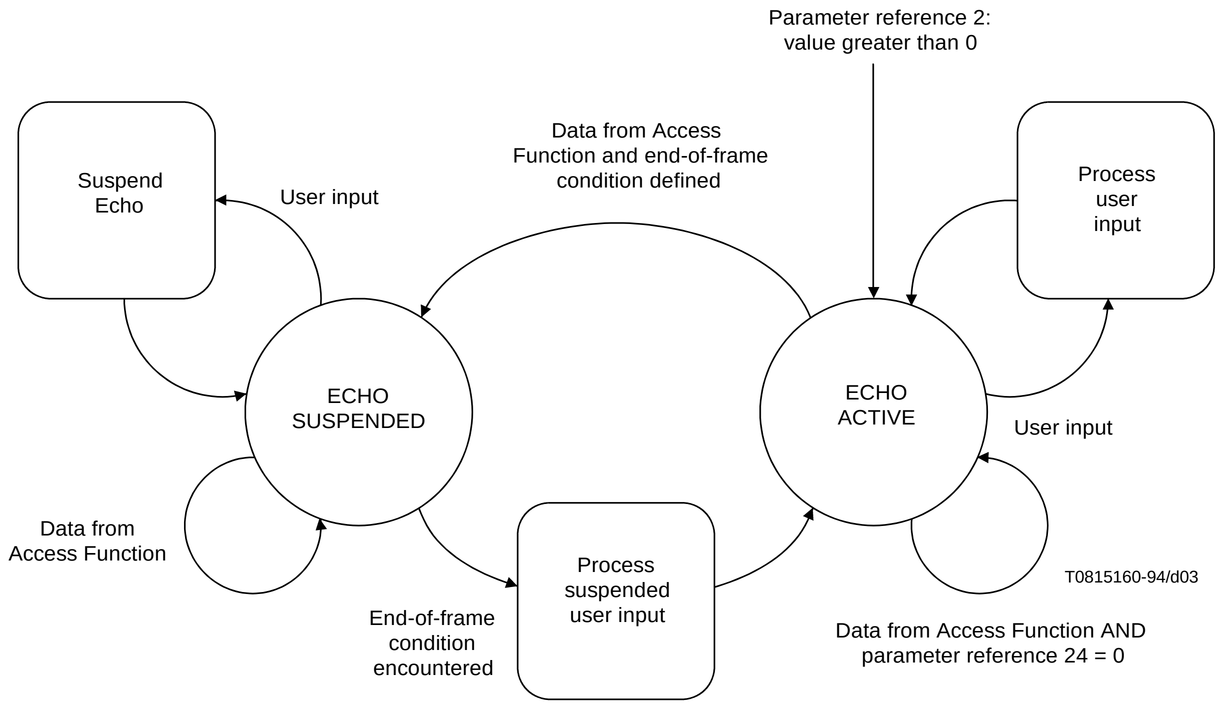
As long as no data have been received from the Access Function, the terminal remains in the state ECHO_ACTIVE.

When the terminal is in the state ECHO_ACTIVE and data are received from the Access Function, one of the following shall apply:

– if an end-of-frame condition has been set (CCITT Recommendation X.3 [3] extended parameter reference 24 is not implemented or set to a non-zero value, see clause 10), the terminal shall enter the state ECHO_SUSPENDED;

– if no end-of-frame condition has been set (parameter reference 24 is set to 0), the terminal shall remain in the state ECHO_ACTIVE.

When the terminal is in the state ECHO_SUSPENDED, data received from the Access Function shall be processed (e.g. displayed) and the echoing of user input shall be suspended until an end-of-frame has been detected.



NOTE – An end-of-frame condition is defined if either CCITT Recommendation X.3 [3] extended parameter reference 24 is not implemented or if parameter reference 24 has been set to a non-zero value.

FIGURE 3/T.105

Echo handler state diagram

When an end-of-frame has been detected in the ECHO_SUSPENDED state, suspended user input shall be processed (e.g. displayed) immediately prior to any other action and the terminal shall return to the ECHO_ACTIVE state.

The possible end-of-frame conditions are:

- the receipt of a complete packet sequence shall be an end-of-frame condition, if CCITT Recommendation X.3 [3] extended parameter reference 24 is not implemented;
- end-of-frame conditions as defined by parameter reference 24;
- the receipt of an SBV PDU different from the SBV_VTX_Data PDU is considered to be an implicit end-of-frame condition.

Certain possibly defined end-of-frame conditions can be checked for by inspecting the contents of the output buffer.

The **forwarding condition handler** manages the forwarding conditions and upon user input forwards data to the Layer 7 protocol machine. It also takes care of the currently defined user input field length (see clause 10).

NOTE 5 – The handling of the input field is a local matter.

The forwarding conditions can be divided into the following groups in a hierarchical order (i.e. lower numbered conditions have precedence):

1) X.29 PAD messages (mandatory);

2) X.3 idle timer expiration (mandatory);

3) X.3 standard forwarding conditions (mandatory);

4) X.3 extended forwarding conditions (optional);

5) definable forwarding conditions (optional).

NOTE 6 – It is a local matter to decide at which point in time the Application Layer data are assembled into X.25 packets. Nevertheless, it is expected that these packets are forwarded whenever a packet is full.

CCITT Recommendation X.29 [6], 2.1, specifies the receipt of a PAD message as a forwarding condition. PAD messages are used to set and read parameters.

If the idle timer is set to values > 0, the contents of the input buffer shall be forwarded if the input buffer has remained unchanged for the period of time as defined by the current value of the idle timer parameter.

CCITT Recommendation X.3 [3] standard forwarding conditions define single octets which trigger forwarding when encountered in the input buffer. The default X.3 forwarding conditions are given in Annex C (normative).

CCITT Recommendation X.3 [3] extended forwarding conditions (see clause 10) define fixed multi-octet sequences, which are in use in Videotex systems, accessed over the PSTN or the PSPDN [see Annex E (informative)].

Definable function keys can be set by the service to define arbitrary forwarding conditions. The support of definable function keys is optional. These conditions are additional to the CCITT Recommendation X.3 [3] forwarding conditions. In cases of an ambiguity between these different sets of forwarding conditions, the CCITT Recommendation X.3 [3] conditions shall have the higher priority.

A terminal supporting definable function keys shall be capable of handling a minimum of 8 function keys with identifications 1 to 8. These function keys may be linked to forwarding conditions. An optional user visible name element may be used to distinguish the functions for the user. The function key definitions may be changed by sending a new definition for some specific key, or deleted by either overwriting them with an empty definition (zero length code sequence) or by resetting all definable function keys.

EXAMPLE: To adapt an ISDN syntax based videotex terminal to a system using function keys INDEX, HELP, DELETE, CORRECT, PREVIOUS, NEXT, REPEAT and SEND in a Videotex Service X, forwarding conditions may be downloaded with key 1 being INDEX, key 2 being HELP, etc. and corresponding codes (e.g. 1/3 x/y). To use this terminal in Videotex Service Y using the same function keys, but other code sequences (e.g. the code sequence for CORRECT shall be 0/8 instead of 1/3 4/7), forwarding conditions are downloaded according to the environment.

A terminal supporting definable function keys may abstain from treating a character sequence as a forwarding condition, if it is entered directly into the input buffer, even though this character sequence is identical to an already defined forwarding condition code sequence linked to a function key. By definition, this rule applies only to definable forwarding conditions as these provide a way to link forwarding conditions to function keys. CCITT Recommendation X.3 [3] forwarding conditions are not subject to this rule. The default state of the list of function keys is specified in Annex C (normative).

NOTE 7 – This rule eases implementations, because otherwise terminals should be capable of interpreting long sequences of input characters.

The **processable data unit** handles processable data as defined in CCITT Recommendation T.101 [8]. It may also be capable of handling processable data transparently without making use of the Dialogue Data Units (DDUs). The default regime to be used for conveying processable data is defined in Annex C (normative).

The **escape functions** handle services which are not specified in this Recommendation. The default states of processes related to SBV_Escape subservices are defined in Annex D (normative).

6.3 Protocol pillar

The protocol pillar to be used to convey the SBV protocol is as given in Figure 4.

7	T.105 T.101 [8] data syntax
6	null (Note 1)
5	null (Note 2)
4	null (Note 2)
3	(Note 3) (Note 4)
2	(Note 3)
1	(Note 3)

Layer

NOTES

1 The main purpose of Layer 6, the conversion from the “abstract syntax” to the “transfer syntax” is not necessary, because in this case the abstract data syntax in Layer 7 is identical to the transfer data syntax. Also all other features of the Layer 6 are not used and therefore “null” is inserted for Layer 6. The abstract syntax in Layer 7 and the coding in Layer 6 correspond to the data syntaxes DS I, DS II and DS III in CCITT Recommendation T.101 [8].

2 For the time being, it is not the intention to include Layers 4 and 5.

3 Not defined by this Recommendation.

4 The upper service interface of Layer 3 shall be in accordance with the BIS as defined in this Recommendation.

FIGURE 4/T.105

Protocol pillar

The application layer protocol (SBV protocol) is defined in this Recommendation.

The data syntax used for syntax-based Videotex is defined in Recommendation T.101 [8].

The lower layer (Layer 1 to Layer 3) protocols are out of the scope of this Recommendation. However, the upper service interface of Layer 3 shall be in accordance with the BIS as defined in clause 11.

NOTE – Recommendation T.106 [12] gives an overview of how the SBV protocol can be conveyed using different types of networks. Lower layer Recommendations for this purpose are also shown in Recommendation T.106 [12].

7 Service definition

7.1 Introduction

The Syntax-based Videotex Application Layer incorporates some functionalities related to those Videotex Services originally defined for the PSTN/PSPDN environment, as well as functionalities related to the additional features as stated in clause 4.

This clause describes, in an abstract way, the services offered by the Syntax-based Videotex Application Layer in terms of:

- the primitive actions and events of the service;
- the parameter data associated with each primitive action and event.

The services are divided into two classes:

- Mandatory Kernel services (see Table 1);
- Optional services (see Table 2).

TABLE 1/T.105

Kernel services (mandatory)

Kernel Service	C/NC	Initiated by TF/AF	Function
SBV_Establish	C	TF	Connection establishment
SBV_Release	NC	Both	Connection release
SBV_Reset	C	AF	Reset to basic state
SBV_VTX_Data	NC	Both	Videotex data transfer
SBV_Set_Param	NC	AF	Set X.3 parameters
SBV_Read_Param	NC	AF	Read X.3 parameters
SBV_Set_Read_Param	NC	AF	Set and read X.3 parameters
SBV_Param_Indication	NC	TF	X.3 parameter indication
SBV_TFI	C	AF (Note)	Exchange of Terminal Facility Identifier (TFI) information
SBV_TC_Error	NC	Both	Telematic Command Error Message

C Confirmed service
NC Non-confirmed service
AF Access Function
TF Terminal Function

NOTE – The support of the SBV_TFI service element shall be mandatory for Syntax-based Videotex Terminals only. It shall be up to the Videotex Service to employ this terminal capability.

For an interim period the implementation of this service element is optional, as long as the TFI is implemented in the terminal and can be transferred in the SBV_VTX_Data service element.

For every service described below, parameters may be mandatory or optional. Mandatory parameters shall always be present within the primitives used to convey the service element and shall be coded as defined in clause 9. Optional parameters need not be present within the primitive used to convey the service element; if an optional parameter is present, it shall be coded according to clause 9; if it is not present, no information shall be associated to this event (i.e. no default value is defined for any parameter).

The protocol is used between a Terminal Function and an Access Function and as such is not symmetrical, because some services shall be invoked by the Terminal Function or the Access Function only.

The default role assignment shall be defined as follows: the **calling** side (i.e. the entity which issues an SBV_Establish Request) shall act as the Terminal Function; the **called** side (i.e. the entity which receives an SBV_Establish Indication) shall act as the Access Function.

NOTE – This role assignment may be reversed upon indication by the caller, but only if the called side accepts (see 7.2.1).

TABLE 2/T.105

Optional services

Telematic Command	C/NC	Initiated by TF/AF	Function
SBV_Channel_Open (Note 1)	C	AF	Invitation to establish an additional communication channel
SBV_Channel_Close (Note 1)	C	AF	Invitation to release an additional communication channel
SBV_Channel-Error (Note 1)	NC	TF	Additional communication channel error message
SBV_Begin_Application	C	TF	Direct selection of a Videotex Application
SBV_End_Application (Note 2)	NC	TF	Return to the Videotex Application selection phase
SBV_End_Immediate (Note 2)	NC	TF	Immediate return to the Videotex Application selection phase
SBV_TPD_Begin (Note 3)	C	Both	Begin of transparent processable data exchange regime
SBV_TPD_End (Note 3)	NC	Both	End of transparent processable data exchange regime
SBV_DFK	NC	AF	Definition of function keys
SBV_Remote_Echo	NC	TF	Invitation to set the remote echo ON or OFF
SBV_Escape	NC	Both	Services not standardized in this Recommendation

C Confirmed service
NC Non-confirmed service
AF Access Function
TF Terminal Function

NOTES

1 If implemented, all three communication channel management services shall be supported.

2 If implemented at AF side, both services shall be supported, except terminals conforming to Recommendation T.105 as per COM VIII-R 47 (1992).

3 If implemented, both Transparent Processable Data (TPD) services shall be supported.

7.2 Kernel services

7.2.1 SBV_Establish

The SBV_Establish service element shall be used to establish a communication. It may be initiated by the Terminal Function only. This service is confirmed, but in case a communication cannot be established this shall be indicated using the SBV_Release service.

A Syntax-based Videotex Terminal shall not be obliged to handle an incoming call.

NOTE 1 – This service allows for a direct selection of a Videotex Application. This direct selection may also be accomplished using the SBV_Begin_Application service, once the communication establishment has been successful.

NOTE 2 – This service may be used to establish several network connections. It is out of the scope for this Recommendation whether to open an additional bearer channel or to establish another network connection on the existing bearer channel. In case of an ISDN a bearer channel is equivalent to a B-channel.

NOTE 3 – A Videotex Service calling an SBV Terminal acts, in terms of the communication model (see 6.1), as a Terminal Function, at least up to that point in time when both sides agree to reverse the role assignment.

The primitives and parameters of the SBV_Establish service are described in Table 3 below.

TABLE 3/T.105

Parameters	Request	Indication	Response	Confirm
OB_Called_Address	Optional	Optional (=)	–	–
OB_Called_Subaddress	Optional	Optional (=)	–	–
OB_Application_Address	Optional	Optional (=)	–	–
OB_Application_Selection	Optional	Optional (=)	–	–
OB_Application_Data	Optional	Optional (=)	–	–
OB_User_Data	Optional	Optional (=)	–	–
IB_Called_Address	Optional	Optional (=)	–	–
IB_Application_Address	Optional	Optional (=)	–	–
IB_Application_Selection	Optional	Optional (=)	–	–
IB_Application_Data	Optional	Optional (=)	–	–
IB_User_Data	Optional	Optional (=)	–	–
Reference	Optional	Optional (=)	–	–
Reverse_Role_Assignment	Optional	Optional (=)	–	–
Result	–	–	Optional	Optional (=)
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request or Response primitive. NOTE – Depending on the network and/or access type it might be necessary to have one or several of these parameters present in a Request in order to establish a communication. However, this is out of the scope of this Recommendation.				

Parameters:

- **OB_Called_Address** – This parameter carries the Out-Band address of the Access Function to be reached.

- **OB_Called_Subaddress** – This parameter carries the Out-Band subaddress of the Access Function; it may only be present if the OB_Called_Address parameter is present.

- **OB_Application_Address** – This parameter carries complementary information about the Videotex Application to be reached; this may include a network address which is interpreted by the Access Function.

- **OB_Application_Selection** – This parameter carries complementary information (e.g. mnemonic) about the Videotex Application to be reached; this is interpreted by the Access Function.

- **OB_Application_Data** – This parameter carries data to be sent to the Videotex Application to be reached; it is passed transparently to the Host Function via the Access Function.

- **OB_User_Data** – This parameter carries user data not specified in this Recommendation which is interpreted by the Access Function.

- **IB_Called_Address** – This parameter carries an In-Band address; it either represents an address of the Videotex Application to be reached in which case it is interpreted by the Access Function or it represents a network address used by the Access Network to identify the called side.

- **IB_Application_Address** – This parameter carries complementary information about the Videotex Application to be reached; this may include a network address; it is interpreted by the Access Function.

- **IB_Application_Selection** – This parameter carries complementary information (e.g. mnemonic) about the Videotex Application to be reached; it is interpreted by the Access Function.

- **IB_Application_Data** – This parameter carries data to be sent to the Videotex Application to be reached; it is passed transparently to the Host Function via the Access Function.

- **IB_User_Data** – This parameter carries user data not specified in this Recommendation which is interpreted by the Access Function.

- **Reference** – This parameter carries a reference value as given by a previously received SBV_Channel_Open Indication. This parameter shall not be present when no reference value is available (see 7.3.1).

- **Reverse_Role_Assignment** – This parameter has no value. When present, it shall act as an indicator that the calling side requests to reverse the default role assignment (i.e. the calling side requests to act as the Access Function instead of the Terminal Function). The inverted role assignment applies upon a "Successful" confirmation. A called side which is not able to support a reversed role assignment may choose to ignore or reject the incoming call.

- **Result** – This parameter carries the result of the communication establishment; it may take the value "Successful" or, in the case of an attempt to establish a communication with a Videotex Application directly via the Access Function, it may take the value "Application not available" if the communication to the Access Function has been successfully established, but the indicated Videotex Application could not be reached.

  NOTE 4 – This parameter may only be present if the CCITT Recommendation X.25 [5] Fast Select facility is used (see 11.3).

  NOTE 5 – In no case this parameter can be used to indicate that the SBV_Establish was entirely not successful. The Access Function will indicate this by using SBV_Release.

7.2.2 SBV_Release

The SBV_Release service shall be used either to release a communication or to indicate that a previous attempt to establish a connection using the SBV_Establish service has not been successful. It may be initiated by both sides at any time. Moreover, in case of an Access Network failure, both sides shall receive an SBV_Release indication. In the case of the release of a "Master"-channel, all the associated "Slave"-channels shall be released (see 6.2).

This service is non-confirmed. The primitives and parameters of the SBV_Release service are described in Table 4 below.

TABLE 4/T.105

Parameters	Request	Indication
OB_Cause	Optional	Optional
IB_Cause	Optional	Optional
IB_Diagnostic	Optional	Optional
Result	Optional	Optional
NOTE – Either OB_Cause or IB_Cause/IB_Diagnostic shall be present in an Indication.		

Parameters:

- **OB_Cause** – This parameter carries information about the reason for the Out-Band release of the communication.

- **IB_Cause** – This parameter carries the reason for the In-Band release of the communication.

- **IB_Diagnostic** – This parameter carries additional information on the reason for the In-Band release of the communication.

- **Result** – This parameter carries the application dependent reason for the release of the communication; it may take the values "Application not available" or "Application already engaged" as the result of a previous attempt to establish a communication with a Videotex Application directly via the Access Function.

  NOTE – This parameter can only appear if the CCITT Recommendation X.25 [5] Fast Select facility is used.

7.2.3 SBV_Reset

The SBV_Reset service element shall be used by the Access Function to reset the Terminal Function into the basic state as defined in Annex C (normative). In addition, all possibly existent "Slave"-channels (see 6.2) related to the communication channel on which the service request was received shall be released. This service is confirmed.

The primitives and parameters of the SBV_Reset service are described in Table 5 below.

TABLE 5/T.105

Parameters	Request	Indication	Response	Confirm
Result	–	–	Conditional C1	Conditional (=) C1
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request or Response primitive.				

Parameters:

- **Result** – This parameter carries the result of the SBV_Reset; it may take the value "Successful" or "Not successful".

  NOTE – As defined by the condition C1, this parameter is used in case of mapping on Telematic Command (see 8.3.3).

The condition C1 depends on the application layer to BIS association (see 8.3.3). The value of the condition C1 is defined by Table 6 below.

TABLE 6/T.105

Mapping of SBV_Reset service on BIS service (Table 22)	C1
SBV_Reset is mapped on a BIS-N-RESET	Excluded
SBV_Reset is mapped on a BIS-N-Q-DATA	Mandatory

7.2.4 SBV_VTX_Data

This service element shall be used to transfer Videotex data. It may be initiated by both sides. This service is non-confirmed. The primitives and parameters of the SBV_VTX_Data service are described in the Table 7 below.

TABLE 7/T.105

Parameters	Request	Indication
VTX_Data	Mandatory	Mandatory (=)
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request primitive.		

Parameters:

- **VTX_Data** – This parameter carries the Videotex data to be transmitted.

7.2.5 X.3 parameter manipulation services

7.2.5.1 SBV_Set_Param

This service element shall be used to set one or more of the CCITT Recommendation X.3 [3] parameters of the Terminal Function. It may be initiated by the Access Function only. This service is non-confirmed. The primitives and parameters of the SBV_Set_Param service are described in Table 8 below.

TABLE 8/T.105

Parameters	Request	Indication
X.3_Parameter_List	Optional	Optional (=)
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request primitive.		

Parameters:

- **X.3_Parameter_List** – This parameter carries the X.3 [3] parameter(s) to be modified.

7.2.5.2 SBV_Read_Param

This service element shall be used to read one or more of the CCITT Recommendation X.3 [3] parameters of the Terminal Function. It may be initiated by the Access Function only. This service is non-confirmed. The primitives and parameters of the SBV_Read_Param service are described in Table 9 below.

TABLE 9/T.105

Parameters	Request	Indication
X.3_Parameter_List	Optional	Optional (=)
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request primitive.		

Parameters:

- **X.3_Parameter_List** – This parameter carries the X.3 parameter(s) to be read.

7.2.5.3 SBV_Set_Read_Param

This service element shall be used to set one or more of the CCITT Recommendation X.3 [3] parameters of the Terminal Function and to read the indicated parameters. It may be initiated by the Access Function only. This service is non-confirmed. The primitives and parameters of the SBV_Set_Read_Param service are described in Table 10 below.

TABLE 10/T.105

Parameters	Request	Indication
X.3_Parameter_List	Optional	Optional (=)
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request primitive.		

Parameters:

- **X.3_Parameter_List** – This parameter carries the X.3 [3] parameter(s) to be modified.

7.2.5.4 SBV_Param_Ind

This service element shall be used to indicate the CCITT Recommendation X.3 [3] parameters values of the Terminal Function. It may be initiated by the Terminal Function only. This service is non-confirmed. The primitives and parameters of the SBV_Param_Ind service are described in Table 11 below.

TABLE 11/T.105

Parameters	Request	Indication
X.3_Parameter_List	Optional	Optional (=)
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request primitive.		

Parameters:

- **X.3_Parameter_List** – This parameter carries the X.3 [3] parameter(s) as defined by CCITT Recommendation X.29 [6].

7.2.6 SBV_TFI

The SBV_TFI service shall be used by the Access Function only to request for the facilities of the Terminal Function. This service is confirmed. The primitives and parameters of the SBV_TFI service are described in Table 12 below.

TABLE 12/T.105

Parameters	Request	Indication	Response	Confirm
TFI_Enq	Mandatory	Mandatory (=)	–	–
TFI_Ack	–	–	Mandatory	Mandatory (=)
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request or Response primitive.				

Parameters:

- **TFI_Enq** – This parameter carries the TFI request data.
- **TFI_Ack** – This parameter carries the TFI response data.

7.2.7 SBV_TC_Error

The SBV_TC_Error service shall be used by the Application Layer itself to indicate to the application that the previously received Telematic Command was erroneous (syntactical error or SBV_Service not supported or inopportune incoming PDU). This service is non-confirmed. The primitives and parameters of the SBV_TC_Error service are described in Table 13 below.

NOTE – As defined in 8.2.7, a TC_Error_PDU is never an inopportune incoming PDU, in the sense of this protocol. It therefore does never cause another TC_Error_PDU, but acts as a (negative) response PDU.

TABLE 13/T.105

Parameters	Request	Indication
Error_Detection_Location	–	Mandatory
Error_Code	–	Mandatory

Parameters:

- **Error_Detection_Location** – This parameter contains the information whether the error has been detected by the local or the remote Application Layer.
- **Error_Code** – This parameter carries information about the reason of the error in the Telematic Command or indicates that the previously received Telematic Command is not supported.

7.3 Optional services

7.3.1 Communication channel management services

The following service elements allows an Access Function to request the Terminal Function to establish or release additional communication channels. If implemented, all communication channel management services shall be supported.

The additional communication channels may be **independent** from or **linked** with the communication channels which carried the request to open them. The distinction between the two types of additional communication channels is possible by using the Reference parameter of the SBV_Channel_Open service:

- **Independent additional communication channel** – No Reference parameter shall be present in the SBV_Channel_Open Request.
- **Linked additional communication channel** – A Reference parameter shall be present in the SBV_Channel_Open service. In that case, the requesting communication channel shall be named a “Master”-channel and the requested communication channel (when opened) shall be named a “Slave”-channel.

The reference value carried in the Reference parameter shall be associated to one and only one opened “Slave”-channel. When the “Slave”-channel has been established by the Terminal Function, the reference value used to establish this communication channel shall be no longer available.

7.3.1.1 SBV_Channel_Open

This service element shall be used by the Access Function to invite the Terminal Function to open an additional communication channel. The opening of the additional communication channel is accomplished by using the SBV_Establish service. The SBV_Channel_Open service is confirmed. The primitives and parameters of the SBV_Channel_Open service are described in Table 14 below.

TABLE 14/T.105

Parameters	Request	Indication	Response	Confirm
Req_OB_Called_Address	Optional	Optional (=)	–	–
Req_OB_Called_Subaddress	Optional	Optional (=)	–	–
OB_Application_Address	Optional	Optional (=)	–	–
OB_Application_Selection	Optional	Optional (=)	–	–
OB_Application_Data	Optional	Optional (=)	–	–
Req_OB_User_Data	Optional	Optional (=)	–	–
Ind_OB_Cause	–	–	Optional	Optional (=)
Req_IB_Called_Address	Optional	Optional (=)	–	–
IB_Application_Address	Optional	Optional (=)	–	–
IB_Application_Selection	Optional	Optional (=)	–	–
IB_Application_Data	Optional	Optional (=)	–	–
Req_IB_User_Data	Optional	Optional (=)	–	–
Packet_Size	Optional	Optional (=)	–	–
Window_Size	Optional	Optional (=)	–	–
Ind_IB_Cause	–	–	Optional	Optional (=)
Ind_IB_Diagnostic	–	–	Optional	Optional (=)
Reference	Optional	Optional (=)	–	–
Result	–	–	Mandatory	Mandatory (=)

(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request or Response primitive.

NOTE – At least one of the following parameters shall be present in the Request:

- Req_OB_Called_Address;
- OB_Application_Address;
- OB_Application_Selection;
- Req_OB_User_Data;
- Req_IB_Called_Address;
- IB_Application_Address;
- IB_Application_Selection;
- Req_IB_User_Data.

Parameters:

- **Req_OB_Called_Address** – Carries the Out-Band address of the Access Function to be reached; it can be mapped onto the OB_Called_Address parameter of the SBV_Establish service which shall be used to open the additional communication channel.

- **Req_OB_Called_Subaddress** – Carries the Out-Band subaddress of the Access Function to be reached; it can be mapped onto the OB_Called_Subaddress parameter of the SBV_Establish service.

- **OB_Application_Address** – Carries complementary information about the Videotex Application to be reached; it can be mapped onto the OB_Application_Address parameter of the SBV_Establish service.

- **OB_Application_Selection** – Carries complementary information (e.g. mnemonic) about the Videotex Application to be reached; it can be mapped onto the OB_Application_Selection parameter of the SBV_Establish.

- **OB_Application_Data** – Carries data to be passed to the Videotex Application to be reached; it can be mapped onto the OB_Application_Data parameter of the SBV_Establish.

- **Req_OB_User_Data** – Carries data to be passed transparently to the requested Application; it can be mapped onto the OB_User_Data parameter of the SBV_Establish service.

- **Ind_OB_Cause** – Carries the reason for the unsuccessful additional Out-Band communication channel establishment; it corresponds to the OB_Cause parameter of the SBV_Release service.

- **Req_IB_Called_Address** – This parameter carries either the address of the Videotex Application or the In-Band address of the Access Function to be reached; it can be mapped onto the IB_Called_Address parameter of the SBV_Establish service which shall be used to open the additional communication channel.

- **IB_Application_Address** – This parameter carries complementary information about the Videotex Application to be reached; it can be mapped onto the IB_Application_Address parameter of the SBV_Establish service.

- **IB_Application_Selection** – Carries complementary information (e.g. mnemonic) about the Videotex Application to be reached; it can be mapped onto the IB_Application_Selection parameter of the SBV_Establish service.

- **IB_Application_Data** – Carries data to be passed to the Videotex Application to be reached; it can be mapped onto the IB_Application_Data parameter of the SBV_Establish service.

- **Req_IB_User_Data** – Carries data to be passed transparently to the requested Application; it can be mapped onto the IB_User_Data parameter of the SBV_Establish service.

- **Packet_Size** – Carries a requested Layer 3 packet size for the establishment of the additional communication channel. If the requested value is not available for any reason, the value which is currently in use on the requesting channel may be employed instead.

- **Window_Size** – Carries a requested Layer 3 window size for the establishment of the additional communication channel. If the requested value is not available for any reason, the value which is currently in use on the requesting channel may be employed instead.

- **Ind_IB_Cause** – Carries the reason for the unsuccessful additional In-Band communication channel establishment; it corresponds to the IB_Cause parameter of the SBV_Release service.

- **Ind_IB_Diagnostic** – Carries additional information about the reason for the unsuccessful additional In-Band communication channel establishment; it corresponds to the IB_Diagnostic parameter of the SBV_Release service.

- **Reference** – Carries a reference information to be sent transparently to the Videotex Application to be reached; this parameter can be mapped onto the Reference parameter of the SBV_Establish service.

- **Result** – Carries the result of the opening of the additional communication channel; it may take the values "Successful", "Not successful" or "Application not available".

If the Result parameter value is "Successful", no Ind_OB_Cause, Ind_IB_Cause or Ind_IB_Diagnostic parameters shall be present.

7.3.1.2 SBV_Channel_Close

The SBV_Channel_Close service element shall be used by the Access Function only to request the release of a previously opened additional "Slave"-channel. Only the request of a channel to release one of its associated "Slave"-channels shall be considered as valid; the release is accomplished by using the SBV_Release service. All other requests shall result in a negative confirmation but shall be otherwise ignored.

This SBV_Channel_Close service is confirmed. The primitives and parameters of the SBV_Channel_Close service are described in Table 15 below.

TABLE 15/T.105

Parameters	Request	Indication	Response	Confirm
Reference	Mandatory	Mandatory (=)	–	–
Result	–	–	Mandatory	Mandatory (=)
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request or Response primitive.				

Parameters:

- **Reference** – This parameter carries the reference of the communication channel to be closed; it takes the value of the Reference parameter of the SBV_Channel_Open service element previously used to request the establishment of the linked additional communication channel.

- **Result** – This parameter carries the result of the release; it may take the value "Successful" if the Release succeeded or "Illegal Reference parameter" if the Reference parameter was erroneous; it may also take the value "Release not allowed" in case of a "Slave"-channel trying to request the closure of its "Master"-channel.

7.3.1.3 SBV_Channel_Error

The SBV_Channel_Error service shall be used by the Terminal Function only to inform the Access Function via the "Master"-channel that a previously opened "Slave"-channel has been disconnected for some reason. This service is non-confirmed. The primitives and parameters of the SBV_Channel_Error service are described in Table 16 below.

TABLE 16/T.105

Parameters	Request	Indication
Reference	Mandatory	Mandatory (=)
Ind_OB_Cause	Optional	Optional (=)
Ind_IB_Cause	Optional	Optional (=)
Ind_IB_Diagnostic	Optional	Optional (=)

(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request primitive.

NOTE – Either Ind_OB_Cause or Ind_IB_Cause/Ind_IB_Diagnostic shall be present in an Indication.

Parameters:

- **Reference** – This parameter carries the reference of the disconnected communication channel; it takes the value of the Reference parameter of the SBV_Channel_Open service element previously used to open the linked additional communication channel.
- **Ind_OB_Cause** – This parameter carries the reason for the Out-Band disconnection of the referenced communication channel.
- **Ind_IB_Cause** – This parameter carries the reason for the In-Band disconnection of the referenced communication channel.
- **Ind_IB_Diagnostic** – This parameter carries additional information on the reason for the In-Band disconnection of the referenced communication channel.

7.3.2 Application selection services

7.3.2.1 SBV_Begin_Application

The SBV_Begin_Application service shall be used by the Terminal Function only to select a Videotex Application directly by short-cutting the dialogue and welcome phase in the Access Function. If a connection to another Videotex Application existed prior to the use of this service, then this connection may be closed (depending on the Videotex Service definition) by the Access Function before trying to establish the requested connection. This service is confirmed.

NOTE – This service does not preclude a direct application selection using the SBV_Establish service.

The primitives and parameters of the SBV_Begin_Application service are described in Table 17 below.

TABLE 17/T.105

Parameters	Request	Indication	Response	Confirm
IB_Application_Address	Optional	Optional (=)	–	–
IB_Application_Selection	Optional	Optional (=)	–	–
IB_Application_Data	Optional	Optional (=)	–	–
IB_User_Data	Optional	Optional (=)	–	–
Result	–	–	Mandatory	Mandatory (=)

(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request or Response primitive.

NOTE – At least one of the following parameters shall be present in the Request:

- IB_Application_Address;
- IB_Application_Selection;
- IB_User_Data.

Parameters:

- I**B_Application_Address** – This parameter carries the address of the Videotex Application to be reached; it is interpreted by the Access Function in order to establish the communication with the requested Application.
- **IB_Application_Selection** – This parameter carries information (e.g. mnemonic) about the Videotex Application to be reached; it is interpreted by the Access Function in order to establish the communication with the requested Application.
- **IB_Application_Data** – This parameter carries data to be passed transparently by the Access Function to the Videotex Application to be reached.
- **IB_User_Data** – This parameter carries data not specified in this Recommendation to be interpreted by the Access Function.
- **Result** – This parameter carries the result of the direct access to the Application; it may take the value "Successful", "Application not available" if the requested Videotex Application is not available or "Application already engaged" for those Access Functions which are not able to disconnect from an engaged application upon reception of an SBV_Begin_Application indication.

7.3.2.2 SBV_End_Application

The SBV_End_Application service shall be used by the Terminal Function only to disconnect from a Host Function and to return to a dialogue and welcome phase. This service is non-confirmed.

Parameters:

None.

7.3.2.3 SBV_End_Immediate

The SBV-End_Immediate service shall be used by the Terminal Function only to disconnect from a Host Function as soon as possible and return to a dialogue and welcome phase. There is a risk of loss of intermediate data. This service is non-confirmed.

Parameters:

None.

7.3.3 Transparent processable data services

The SBV_TPD (Transparent Processable Data) services may be used to allow for the exchange of processable data (see Rec. T.101 [8], DS II) transparently without making use of the Dialogue Data Units (DDUs). If implemented, both transparent processable data services (SBV_TPD_Begin and SBV_TPD_End) shall be supported.

7.3.3.1 SBV_TPD_Begin

The SBV_TPD_Begin service shall be used to request for the exchange of transparent processable data. It may be initiated by both sides. This service is confirmed. The primitives and parameters of the SBV_TPD_Begin service are described in Table 18 below.

TABLE 18/T.105

Parameters	Request	Indication	Response	Confirm
Result	–	–	Mandatory	Mandatory (=)
DDU_Fall-Back_Mode	–	–	Optional	Optional (=)
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request or Response primitive.				

Parameters:

- **Result** – This parameter carries the information whether the receiving side shall support the transparent transmission of processable data or not. It may take the values “Successful” or “TPD not supported”.

  If the Result parameter takes the value “Successful”, then the processable data shall be exchanged transparently and the following application rules shall apply:

  - The DDU Layer shall not be used.
  - A T-Associate request shall be issued immediately upon the receipt of an SBV_TPD_Begin Confirmation indicating “Successful”.
  - A TDU is carried in one single complete packet sequence of DATA packets with the Q-bit set to 0.
  - A complete packet sequence may contain one and only one TDU.
  - 8-bit processable data are exchanged via DATA packets with the Q-bit set to 0.
  - The regime of an SBV_TPD_Begin lasts up to the transmission of an SBV_TPD_End Request.

  If the Result parameter takes the value “TPD not supported”, then the processable data shall be exchanged as defined in Recommendation T.101 [8], DS II using the SBV_VTX_Data service, if possible. The regime of transparent processable data exchange shall be implicitly terminated.

- **DDU_Fall-Back_Mode** – This parameter may be present if the value of the Result parameter indicates “TPD not supported”. It may give additional information about supported DDU modes on the receiving side. This parameter may take values from “A” to “G” and their permitted combinations (see 9.3.3.2), corresponding to the respective DDU modes as specified in Recommendation T.101 [8].

7.3.3.2 SBV_TPD_End

The SBV_TPD_End service shall be used to end the regime of a previously established transparent processable data exchange. This service is non-confirmed.

The receipt of an SBV_TPD_End indication shall release any possibly pending T-Associate regime in the sense of Recommendation T.101 [8].

Parameters:

None.

7.3.4 SBV_DFK

The SBV_DFK (Definable Function Keys) service shall be used to define function keys. It shall be initiated by the Access Function only. This service is non-confirmed. The primitives and parameters of the SBV_DFK service are described in Table 19 below.

TABLE 19/T.105

Parameters	Request	Indication
Function_Keys	Optional	Optional (=)
Reset_Keys	Optional	Optional (=)
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request primitive.		

Parameters:

At least one of the parameters should be used. If no parameter is present in an SBV_DFK Request, the service primitive shall be ignored.

- **Reset_Keys** – This parameter indicates that all definable function keys set by a previous SBV_DFK invocation are to be cleared. Forwarding conditions set by CCITT Recommendation X.3 [3] set commands are not influenced by this reset mechanism.
- **Function_Keys** – This parameter carries a list of function key definitions. Each function key definition consists of an identification, a user-visible name, an indication whether pressing the function key shall be a forwarding condition, and a code sequence (a sequence of octets which has to be sent to the Access Function when the forwarding condition occurs).

7.3.5 SBV_Remote_Echo

The SBV_Remote_Echo shall be used by the Terminal-Function only to request the Access Function to stop or to start the echo. This service is non-confirmed. The primitive and parameter of the SBV_Remote_Echo service are described in Table 20 below.

NOTE – In some cases the AF may not be able to fulfil the request (e.g. when waiting for a user to select a new page, the AF might not be prepared to switch echo off).

TABLE 20/T.105

Parameters	Request	Indication
Echo	Mandatory	Mandatory (=)

Parameters:

- **Echo** – This parameter carries information about the echo management. It may take the value "ON" or "OFF".

7.3.6 SBV_Escape

The SBV_Escape service is used for functions which exist in some Videotex Services (e.g. bulk update, transport of binary data). For details concerning this service refer to Annex D (informative). This service is non-confirmed. The primitives and parameters of the SBV_Escape service are described in Table 21 below.

TABLE 21/T.105

Parameters	Request	Indication
Service_ID	Mandatory	Mandatory (=)
Data	Optional	Optional
(=) The value of this parameter is identical to the one of the corresponding parameter in the preceding Request primitive.		

Parameters:

- **Service-ID** – This parameter carries the identification of a specific service [see 9.1 and Annex D (informative)].
- **Data –** This parameter carries data not standardized in this Recommendation.

8 Protocol

8.1 Association of SBV Application Layer and BIS primitives

Table 22 shows the association between SBV application layer PDUs and BIS primitives (see clause 11). For each application layer PDU the corresponding BIS primitive is given.

TABLE 22/T.105

Association of SBV application layer and BIS

SBV Application Layer PDU	SBV_NL Primitives	Allowed primitive type	
		Req./Ind.	Rsp./Con.
SBV_Establish	BIS-N-CONNECT	X	X
SBV_Release	BIS-N-DISCONNECT	X	
SBV_Reset	BIS-N-RESET or BIS-N-Q-DATA (Note)	X X	X X
SBV_VTX_Data	BIS-N-DATA	X	
SBV_Set_Param	BIS-N-Q-DATA	X	
SBV_Read_Param	BIS-N-Q-DATA	X	
SBV_Set_Read_Param	BIS-N-Q-DATA	X	
SBV_Param_Ind	BIS-N-Q-DATA	X	
SBV_TFI	BIS-N-Q-DATA	X	X
SBV_TC_Error	BIS-N-Q-DATA	X	
SBV_Channel_Open	BIS-N-Q-DATA	X	X
SBV_Channel_Close	BIS-N-Q-DATA	X	X
SBV_Channel_Error	BIS-N-Q-DATA	X	
SBV_Begin_Application	BIS-N-Q-DATA	X	X
SBV_End_Application	BIS-N-Q-DATA	X	
SBV_End_Immediate	BIS-N-INTERRUPT	X	
SBV_TPD_Begin	BIS-N-Q-DATA	X	X
SBV_TPD_End	BIS-N-Q-DATA	X	
SBV_DFK	BIS-N-Q-DATA	X	
SBV_Remote_Echo	BIS-N-Q-DATA	X	
SBV_Escape	BIS-N-Q-DATA	X	X

Req. Request
Ind. Indication
Rsp. Response
Con. Confirmation

NOTE – See 8.2.3.

8.2 Procedures

Except for the SBV_Establish and the SBV_Release service, all Application Layer service primitives shall be issued only on an established network connection.

On a given network connection, all confirmed services shall proceed in accordance with the following rules:

- After having issued a Request-PDU, the application layer shall not issue any other PDU (except for the SBV_Release-Request-PDU or the SBV_Reset-Request-PDU or the TC_Error-PDU) before the reception of the Response-PDU.
- Upon reception of a Request-PDU, the application layer shall not issue any other PDU (except for the SBV_Release-Request-PDU or the SBV_Reset-Request-PDU) than the Response-PDU to the requested service.
- After the reception of a Request-PDU, any reception of another Request-PDU (except for the SBV_Release-Request-PDU or the SBV_Reset-Request-PDU) before the issuing of the Response shall be treated as an SBV-protocol error.
- An SBV-protocol error detected by the application layer will cause a TC_Error-PDU send by the application layer to the remote application layer. This TC_Error-PDU replaces the outstanding Response-PDU, which shall not be issued by the application layer even if the Response Primitive already occurred at the upper service interface.
- After having issued a Request-PDU, the reception of either an SBV_Release-Request-PDU or an SBV_Reset-Request-PDU or a TC_Error-PDU implicitly results in a (negative) confirmation.
- At the issuing time of confirmed service Request-PDU, a timer shall be started; when the timer runs out and the Response-PDU has not been received, the communication shall be released.

  The initial value of this timer shall not be smaller than timer value T28 multiplied by counter value R28 as defined in ISO 8208 [15].

NOTE – It is a local matter and therefore out of scope for the protocol definition, how the requirements given above are fulfilled locally, e.g. with respect to the upper service interface, its primitives and its user application.

CCITT Recommendation X.29 [6] related protocol errors (e.g. unrecognizable X.29 messages) shall be treated in accordance to CCITT Recommendation X.29 [6].

Protocol errors related to Telematic Commands shall be treated by the protocol as defined in this Recommendation.

8.2.1 SBV_Establish

No additional procedures defined.

NOTE – As there is no “negative” SBV_Establish response PDU, an SBV_Release request PDU will occur when the communication cannot be established.

8.2.2 SBV_Release

No additional procedures defined.

8.2.3 SBV_Reset

A terminal function shall perform the SBV_Reset service when receiving either a DTE originated BIS_N_RESET indication or an SBV_Reset_TC or both.

NOTE – Terminals conforming to T.105 as per COM VIII-R 47 (1992) react to an SBV_Reset_TC by sending an SBV_Error_TC.

8.2.4 SBV_VTX_Data

No additional procedures defined.

8.2.5 X.3 parameter manipulation services

The detailed procedures to set and/or read one or more of the available CCITT Recommendation X.3 [3] parameters or extended parameters (see clause 10) of the Terminal Function are described in the CCITT Recommendation X.29 [6], 1.5, 3.1 and 3.5.1.

Protocol errors related to these services (e.g. erroneous X.29 PAD messages) shall be treated in accordance to CCITT Recommendation X.29 [6].

8.2.5.1 SBV_Set_Param

No additional procedures defined.

8.2.5.2 SBV_Read_Param

No additional procedures defined.

8.2.5.3 SBV_Set_Read_Param

No additional procedures defined.

8.2.5.4 SBV_Param_Ind

No additional procedures defined.

8.2.6 SBV_TFI

No additional procedures defined.

8.2.7 SBV_TC_Error

An SBV_TC_Error PDU is sent by the Application Layer when a previously received Telematic Command PDU was either not supported or erroneous. This does not apply for any previously received TC_Error PDU (regardless whether their parameters are valid or not). At the same time, a TC_Error_Indication is given to the local application with the Error_Detection_Location parameter set to “local”.

A valid incoming SBV_TC_Error_TC PDU results in an SBV_TC_Error Indication primitive with the Error_Detection_Location parameter set to “remote”.

8.2.8 Communication channel management services

The procedures below apply only if the communication channel management services are implemented.

8.2.8.1 SBV_Channel_Open

Upon reception of a valid SBV_Channel_Open request PDU, the Terminal Function doubles the instance (see Figure 2) and transmits an SBV_Establish request PDU on the new instance. After the reception of SBV_Establish response PDU or an SBV_Release PDU, the Terminal Function shall issue the SBV_Channel_Open response PDU which shall reflect the result of the attempted connection establishment.

8.2.8.2 SBV_Channel_Close

Upon reception a valid SBV_Channel_Close request PDU, the contained Reference parameter shall be inspected.

The forwarded reference value shall be considered as invalid, if:

- there is no communication channel associated to the reference value;
- the associated communication channel is not a Slave-channel in relation to the communication channel on which the SBV_Channel_Close indication has been received.

If the reference value is invalid, the Terminal Function issues an SBV_Channel_Close response PDU with the Result parameter set accordingly.

If the Reference parameter is valid, the Terminal Function shall transmit an SBV_Release request PDU on the network connection associated to the Reference parameter. The Terminal Function shall then issue an SBV_Channel_Close response PDU positive with the appropriate value in the Result parameter.

8.2.8.3 SBV_Channel_Error

No additional procedures defined.

8.2.9 Application selection services

8.2.9.1 SBV_Begin_Application

No additional procedures defined.

8.2.9.2 SBV_End_Application

No additional procedures defined.

8.2.9.3 SBV_End_Immediate

No additional procedures defined.

8.2.10 Transparent processable data services

8.2.10.1 SBV_TPD_Begin

If the Result parameter in an SBV_TPD_Begin response PDU is set to “TPD not supported” the regime for the exchange of transparent processable data is implicitly terminated.

8.2.10.2 SBV_TPD_End

The reception of an SBV_TPD_End_TC request PDU terminates the regime for the exchange of transparent processable data. An SBV_TPD_End_TC request PDU may occur even if there is no Regime activated by the SBV_TPD_Begin Service used recently.

8.2.11 SBV_DFK

The reception of an otherwise valid SBV_DFK request PDU which contains a reference to a function key which is not supported shall result in the transmission of an SBV_TC_Error response PDU indicating “TC erroneous”.

8.2.12 SBV_Remote_Echo

No additional procedures defined.

8.2.13 SBV_Escape

No additional procedures defined.

8.3 Mapping of parameters

The following Tables 23 to 44 contain the mappings between the parameters of the higher layer service primitives and the parameters of the BIS primitives.

An Application Layer parameter may not be applicable to all forms of a primitive (request, indication, response and confirm). The applicability of the parameters is defined in clause 7 and not repeated here.

The X.3 parameter manipulating services and the Telematic Commands (SBV_TFI to SBV_Escape) make use of only one SBV-IB service primitive, namely BIS-N-Q-DATA. At the higher layers, a discrimination is possible by inspecting the contents of the BIS user data parameter.

NOTE – The Telematic Commands have assigned a Telematic Command header (see 9.1). It precedes the actual parameters of a Telematic Command, and is used to distinguish between the different Telematic Command service primitives. Although this header is not defined as a parameter at the higher layers, it nevertheless appears in the tables below, because it is carried in the user data parameter of the BIS-N-Q-DATA service.

8.3.1 SBV_Establish

See Table 23.

8.3.2 SBV_Release

See Table 24.

TABLE 23/T.105

Parameter associations for the SBV_Establish primitive

SBV_Establish parameters	BIS-N-CONNECT parameters
OB_Called_Address	Called/Responding Address
OB_Called_Subaddress	Called/Responding Address
not used	Calling Address (Note 1)
OB_Application_Address	BIS-user-data
OB_Application_Selection	BIS-user-data
OB_Application_Data	BIS-user-data
OB_User_Data	BIS-user-data
IB_Called_Address	Called/Responding Address or BIS-user-data (Note 2)
IB_Application_Address	BIS-user-data
IB_Application_Selection	BIS-user-data
IB_Application_Data	BIS-user-data
IB_User_Data	BIS-user-data
Reference	BIS-user-data
Reverse_Role_Assignment	BIS-user-data
Result	BIS-user-data (Note 3)

NOTES

1 This parameter is not processed by the Application Layer.

2 When In-Band Addressing is used (e.g. Recommendation X.31, case A), the SBV_Establish parameter IB_Called_Address is mapped onto the Called/Responding Address parameter of the BIS-N-CONNECT primitive. Otherwise, it is mapped onto BIS-user-data parameter of the BIS-N-CONNECT primitive. Note that the mapping of those two BIS-N-CONNECT parameters onto the same X.25 field in both cases does not contradict this way of description.

3 This parameter can only be used in conjunction with the Recommendation X.25 Fast Select facility.

TABLE 24/T.105

Parameter associations for the SBV_Release primitive

SBV_Release parameters	BIS-N-DISCONNECT parameters
OB_Cause	Originator, Reason (Note 1)
IB_Cause	Originator (Note 2)
IB_Diagnostic	Reason (Note 2)
Result	BIS-user-data (Note 3)

NOTES

1 When BIS parameters (Originator, Reason) take Out-Band related values.

2 When BIS parameters (Originator, Reason) take In-Band related values.

3 This parameter can only be used in conjunction with the X.25 Fast Select facility.

8.3.3 SBV_Reset

Depending on the application layer to BIS association (see Table 22), the mapping of Table 25 or Table 26 applies.

TABLE 25/T.105

Parameter associations for the SBV_Reset (N-Reset) primitive (N-Reset)

SBV_Reset parameters	BIS-N-RESET parameters
Not used Not used	Originator (Note) Reason (Note)
NOTE – These parameters are not processed by the Application Layer.	

TABLE 26/T.105

Parameter associations for the SBV_Reset primitive (Telematic Command)

SBV_Reset parameters	BIS-N-Q-DATA parameters
Telematic command header Result	BIS-user-data BIS-user-data

8.3.4 SBV_VTX_Data

See Table 27.

TABLE 27/T.105

Parameter associations for the SBV_VTX_Data primitive

SBV_VTX_Data parameter	BIS-N-DATA parameter
VTX_DATA	BIS-user-data

8.3.5 X.3 parameter manipulation services

8.3.5.1 SBV_Set_Param

See Table 28.

TABLE 28/T.105

Parameter associations for the SBV_Set_Param primitive

SBV_Set_Param parameter	BIS-N-Q-DATA parameter
X.3_Parameter_List	BIS-user-data

8.3.5.2 SBV_Read_Param

See Table 29.

TABLE 29/T.105

Parameter associations for the SBV_Read_Param primitive

SBV_Read_Param parameter	BIS-N-Q-DATA parameter
X.3_Parameter_List	BIS-user-data

8.3.5.3 SBV_Set_Read_Param

See Table 30.

TABLE 30/T.105

Parameter associations for the SBV_Set_Read_Param primitive

SBV_Set_Read_Param parameter	BIS-N-Q-DATA parameter
X.3_Parameter_List	BIS-user-data

8.3.5.4 SBV_Param_Ind

See Table 31.

TABLE 31/T.105

Parameter associations for the SBV_Param_Ind primitive

SBV_Param_Ind parameter	BIS-N-Q-DATA parameter
X.3_Parameter_List	BIS-user-data

8.3.6 SBV_TFI

See Table 32.

TABLE 32/T.105

Parameter associations for the SBV_TFI primitive

SBV_TFI parameters	BIS-N-Q-DATA parameters
Telematic command header TFI_Enq TFI_Ack	BIS-user-data BIS-user-data BIS-user-data

8.3.7 SBV_TC_Error

See Table 33.

TABLE 33/T.105

Parameter associations for the SBV_TC_Error primitive

SBV_TC_Error parameters	BIS-N-Q-DATA parameters
Telematic command header Error_Code	BIS-user-data BIS-user-data
NOTE – The Error_Detection_Location parameter is local.	

8.3.8 Communication channel management services

8.3.8.1 SBV_Channel_Open

See Table 34.

TABLE 34/T.105

Parameter associations for SBV_Channel_Open primitive

SBV_Channel_Open parameters	BIS-N-Q-DATA parameters
Telematic command header (TCH)	BIS-user-data
Req_OB_Called_Address	BIS-user-data
Req_OB_Called_Subaddress	BIS-user-data
OB_Application_Address	BIS-user-data
OB_Application_Selection	BIS-user-data
OB_Application_Data	BIS-user-data
Req_OB_User_Data	BIS-user-data
Ind_OB_Cause	BIS-user-data
Req_IB_Called_Address	BIS-user-data
IB_Application_Address	BIS-user-data
IB_Application_Selection	BIS-user-data
IB_Application_Data	BIS-user-data
Req_IB_User_Data	BIS-user-data
Packet_Size	BIS-user-data
Window_Size	BIS-user-data
Ind_IB_Cause	BIS-user-data
Ind_IB_Diagnostic	BIS-user-data
Reference	BIS-user-data
Result	BIS-user-data

8.3.8.2 SBV_Channel_Close

See Table 35.

TABLE 35/T.105

Parameter associations for the SBV_Channel_Close primitive

SBV_Channel_Close parameters	BIS-N-Q-DATA parameters
Telematic command header Reference Result	BIS-user-data BIS-user-data BIS-user-data

8.3.8.3 SBV_Channel_Error

See Table 36.

TABLE 36/T.105

Parameter associations for the SBV_Channel_Error primitive

SBV_Channel_Error parameters	BIS-N-Q-DATA parameters
Telematic command header Reference Ind_OB_Cause Ind_IB_Cause Ind_IB_Diagnostic	BIS-user-data BIS-user-data BIS-user-data BIS-user-data BIS-user-data

8.3.9 Application selection services

8.3.9.1 SBV_Begin_Application

See Table 37.

TABLE 37/T.105

Parameter associations for the SBV_Begin_Application primitive

SBV_Begin_Application parameters	BIS-N-Q-DATA parameters
Telematic command header IB_Application_Address IB_Application_Selection IB_Application_Data IB_User_Data Result	BIS-user-data BIS-user-data BIS-user-data BIS-user-data BIS-user-data BIS-user-data

8.3.9.2 SBV_End_Application

See Table 38.

TABLE 38/T.105

Parameter associations for the SBV_End_Application primitive

SBV_End_Application parameter	BIS-N-Q-DATA parameter
Telematic command header	BIS-user-data

8.3.9.3 SBV_End_Immediate

See Table 39.

TABLE 39/T.105

Parameter associations for the SBV_End_Immediate primitive

SBV_End_Immediate parameter	BIS-N-INTERRUPT parameter
Not used	BIS-Interrupt-user-data (Value 08/00)
NOTE – The SBV_End_Immediate is mapped upon the BIS-N-INTERRUPT primitive with BIS-Interrupt-user-data containing one single byte with the value 08/00.	

8.3.10 Transparent processable data services

8.3.10.1 SBV_TPD_Begin

See Table 40.

TABLE 40/T.105

Parameter associations for the SBV_TPD_Begin primitive

SBV_TPD_Begin parameters	BIS-N-Q-DATA parameters
Telematic command header Result DDU_Fall_Back_Mode	BIS-user-data BIS-user-data BIS-user-data

8.3.10.2 SBV_TPD_End

See Table 41.

TABLE 41/T.105

Parameter associations for the SBV_TPD_End primitive

SBV_TPD_End parameter	BIS-N-Q-DATA parameter
Telematic command header	BIS-user-data

8.3.11 SBV_DFK

See Table 42.

TABLE 42/T.105

Parameter associations for the SBV_DFK primitive

SBV_DFK parameters	BIS-N-Q-DATA parameters
Telematic command header Function_Keys Reset_Keys	BIS-user-data BIS-user-data BIS-user-data

8.3.12 SBV_Remote_Echo

See Table 43.

TABLE 43/T.105

Parameter associations for the SBV_Remote_Echo primitive

SBV_Remote_Echo parameters	BIS-N-Q-DATA parameters
Telematic command header Echo	BIS-user-data BIS-user-data

8.3.13 SBV_Escape

See Table 44.

TABLE 44/T.105

Parameter associations for the SBV_Escape primitive

SBV_Escape parameters	BIS-N-Q-DATA parameters
Telematic command header Service-ID (Note) Data	BIS-user-data BIS-user-data BIS-user-data
NOTE – The Service-ID parameter corresponds to the third octet of the telematic command header (see 9.1).	

9 Coding

9.1 Coding of the Telematic Command PDUs

Telematic Command PDUs are carried in the BIS user data parameter of the BIS-N-Q-DATA primitive. The Telematic Command PDU consists of a Telematic command header (TCH) of four bytes, immediately followed by the Telematic command message (TCM).

The first octet carried of the TCH shall be coded 04/00.

NOTE 1 – This value is described in CCITT Recommendation X.29 (92), 4.4.1, and indicates “Telematic service message, Videotex”.

The second octet of the TCH shall be coded 09/14, indicating “Syntax-based Videotex Telematic Command”.

NOTE 2 – Both the protocol defined in this Recommendation and Rec. T.101 [8] make use of CCITT Recommendation X.29 (92) extension “Telematic service message, Videotex”. The value 09/14 corresponds to “tag [30]” as reserved for the Syntax-based Videotex Terminal protocol.

The third octet shall contain the codepoint identifying the Telematic Command. The range of the Telematic Command code values carried in this octet is divided into three parts:

00/00-07/14: used for services defined in this Recommendation;

08/00-11/14: used for SBV_Escape subservices;

12/00-15/15: for private use.

Both values 07/15 and 11/15 are reserved for future extensions.

The Telematic Command code values are given in Table 45 below.

TABLE 45/T.105

Telematic Command code values

Telematic Command (Note)	Telematic Command code
SBV_TC_Error_TC	00/01
SBV_Channel_Open_TC	00/02
SBV_Channel_Close_TC	00/03
SBV_Channel_Error_TC	00/04
SBV_TFI_TC	00/05
SBV_Begin_Application_TC	00/06
SBV_End_Application_TC	00/07
SBV_TPD_Begin_TC	00/08
SBV_TPD_End_TC	00/09
SBV_DFK_TC	00/10
SBV_Reset_TC	00/11
SBV_Remote_Echo_TC	00/12
NOTE – Octet 3 can directly indicate the purpose of the command. In that case, it corresponds to the Service-ID parameter of the SBV_Escape service [see Annex D (informative)].	

The fourth octet carries the total length of the subsequent Telematic Command Message. This length may be in the range from 0 (00/00) to 254 (15/14). The value 255 (15/15) is reserved for future extensions.

The TCM shall start on the fifth octet of the TC PDU. It carries the sub-fields associated to the parameters of the Telematic Command. For certain Telematic Commands, the TCM may be empty.

A summary of the structure of the TC PDU is given in Figure 5.

Octet 1	04/00
Octet 2	09/14
Octet 3	TC code
Octet 4	TC length
Octet 5 . . .	TC message . . .

FIGURE 5/T.105

Coding and structure of a Telematic Command PDU

9.2 Coding structures

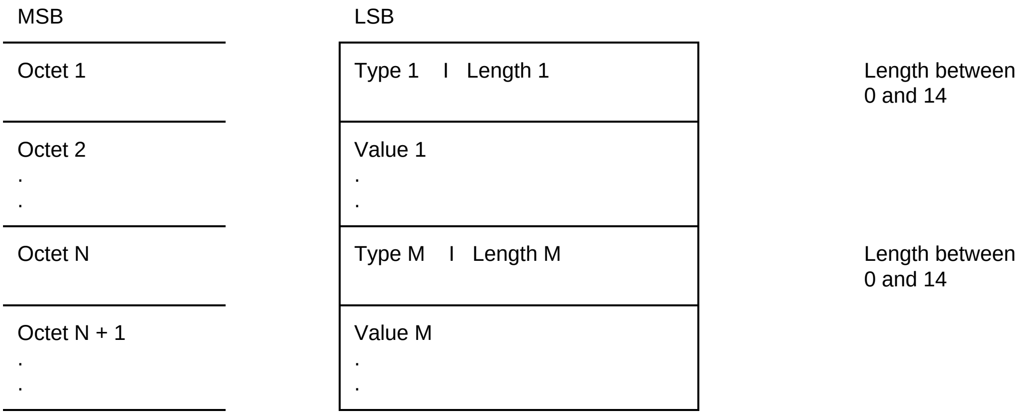
The coding of the various sub-fields associated to the parameters given in Table 46 (see 9.3.1) uses a Type, Length, Value (TLV) method. This coding scheme offers a basic structure and an extended structure. It is applicable to parameters which are carried in the user data fields of the BIS-N-CONNECT and BIS-N-DISCONNECT services and to parameters carried in a Telematic Command.

NOTE – The Basic Coding Structure is necessary due to the limited amount of data which can be carried in the user data fields of the BIS-N-CONNECT service when the X.25 Fast Select facility is not used. However, to encode the different parameters carried in one service element, it may be necessary to employ both coding methods, respectively.

When applying this coding scheme to parameters which are mapped to the user data field of the BIS-N-CONNECT service (regardless whether the X.25 Fast Select facility is used or not), the coding shall start on the fifth octet of the user data field. CCITT Recommendation X.29 [6] reserves the first four octets for the protocol identifier.

9.2.1 Basic coding structure

In this coding structure, the type indicator and the length indicator are coded on 4 bits, respectively. The value of the length indicates the length in octets of the following field. Figure 6 gives an overview:



MSB	LSB	
Octet 1	Type 1 I Length 1	Length between 0 and 14
Octet 2 . .	Value 1 . .	
Octet N	Type M I Length M	Length between 0 and 14
Octet N + 1 . .	Value M . .	

FIGURE 6/T.105

Basic coding structure

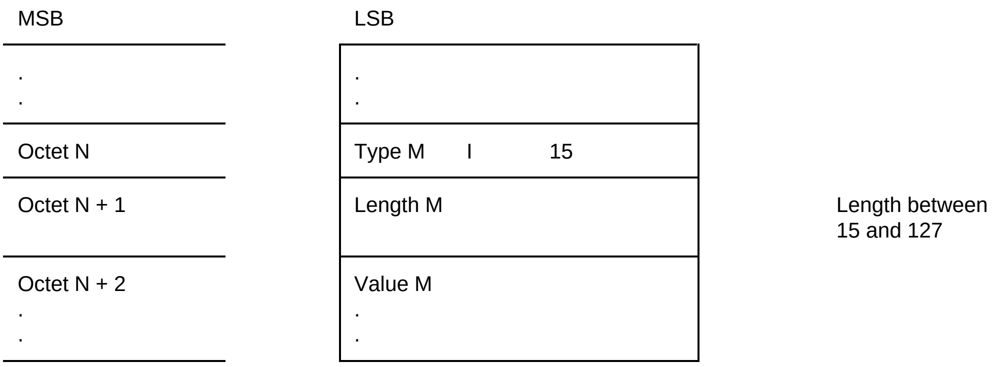
The type indicator may take the values:

- 0-7: not defined in this Recommendation;
- 8-14: defined in this Recommendation;
- 15: reserved for the extended coding structure.

The length indicator may take values between 0 and 14.

A value of 15 in the length indicator implies that the following octet represents the actual length of the parameter value. This actual length shall take a value in the range from 15 (00/15) to 127 (07/15). Values greater than 127 (07/15) are reserved for future extensions.

Figure 7 predicts the coding scheme for that case.



MSB	LSB	
. .	. .	
Octet N	Type M I 15	
Octet N + 1	Length M	Length between 15 and 127
Octet N + 2 . .	Value M . .	

FIGURE 7/T.105

Basic coding structure with length extension mechanism

9.2.2 Extended coding structure

If the first octet of a parameter carries the value 255 (15/15), the following octets shall contain an extended type indicator coded in one octet and a length indicator coded in one octet, as presented in Figure 8.

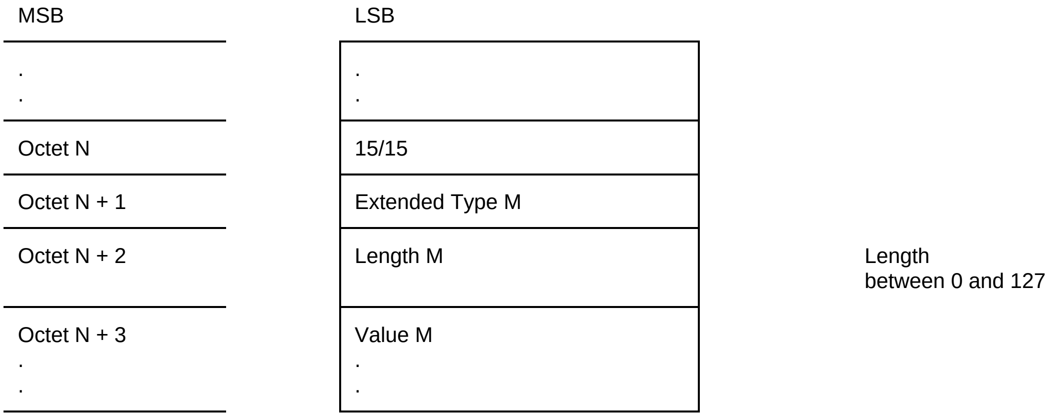


FIGURE 8/T.105

Extended coding structure

The extended type indicator may take the values:

00/00-07/15: not defined in this Recommendation;

08/00-15/15: defined in this Recommendation.

NOTE – In order to avoid any ambiguities between the basic and the extended coding structures, type indicator values x/00 (x in the range from 8 to 15) are reserved for use with the basic coding structure.

In the extended coding structure, the length indicator may take a value in the range from 0 (00/00) to 127 (07/15). Values greater than 127 are reserved for future extensions.

9.2.3 Coding of data types

9.2.3.1 Integers

Integer values shall be coded in one octet. The contents of this octet shall be the binary number equal to the integer value.

NOTE – Integers coded in more than one octet are for further study.

9.2.3.2 Strings

String values shall be coded as a sequence of characters as defined in CCITT Recommendation T.51 [2].

9.2.3.3 Voids

The void data type shall be coded having zero-length and no value.

9.2.3.4 Octet sequences

The octet sequence shall be coded as a sequence of arbitrary octets.

9.3 Coding of parameters

The coding of the following parameters is not performed by the Application Layer, but shall be performed by the BIS:

- OB_Called_Address;
- OB_Called_Subaddress;
- OB_Cause;
- IB_Cause;
- IB_Diagnostic.

Depending on whether In-Band addressing is used or not, the coding of the parameter

- IB_Called_Address

is performed by either the BIS (with In-Band addressing) or the Application layer (no In-Band addressing). In the latter case, the coding shall be as defined by ISO 8208 [15], subclauses 12.2.1.1.2 and 12.2.1.1.3.

9.3.1 Coding of user data fields using the coding structures

Table 46 defines the coding of parameters which are carried in the user data fields of the BIS-N-CONNECT and BIS-N-DISCONNECT services and to parameters carried in a Telematic Command PDU, using the coding structure as defined in 9.2.

TABLE 46/T.105

Parameter	PDU(s)	Type BCS	Type ECS	Value or Remarks
DDU_Fall-Back_Mode	SBV_TPD_Begin_TC	–	12/02	String (9.3.3.2)
Error_Code	SBV_TC_Error_TC	–	08/01	Octet sequence (9.3.3.1)
Function_Keys	SBV_DFK_TC	–	14/01	Octet sequence (9.3.3.3)
IB_Application_Address	BIS-N-CONNECT SBV_Channel_Open_TC SBV_Begin_Application_TC	9	–	(Note 1)
IB_Application_Data	BIS-N-CONNECT SBV_Channel_Open_TC SBV_Begin_Application_TC	11	–	Not defined in this Recommendation
IB_Application_Selection	BIS-N-CONNECT SBV_Channel_Open_TC SBV_Begin_Application_TC	10	–	Not defined in this Recommendation
IB_User_Data	BIS-N-CONNECT SBV_Begin_Application_TC			(Note 2)
Ind_IB_Cause	SBV_Channel_Open_TC SBV_Channel_Error_TC	–	08/03	Not defined in this Recommendation
Ind_IB_Diagnostic	SBV_Channel_Open_TC SBV_Channel_Error_TC	–	08/05	Not defined in this Recommendation
Ind_OB_Cause	SBV_Channel_Open_TC SBV_Channel_Error_TC	–	08/11	Not defined in this Recommendation
OB_Application_Address	BIS-N-CONNECT SBV_Channel_Open_TC	–	09/08	(Note 1)
OB_Application_Data	BIS-N-CONNECT SBV_Channel_Open_TC	–	11/08	Not defined in this Recommendation
OB_Application_Selection	BIS-N-CONNECT SBV_Channel_Open_TC	–	10/08	Not defined in this Recommendation
OB_User_Data	BIS-N-CONNECT			(Note 2)

TABLE 46/T.105 *(end)*

Parameter	PDU(s)	Type BCS	Type ECS	Value or Remarks
Packet_Size	SBV_Channel_Open_TC	–	13/01	Integer (Note 3) 7 ::= 128 octets 8 ::= 256 octets 9 ::= 512 octets 10 ::= 1024 octets 11 ::= 2048 octets
Window_Size	SBV_Channel_Open_TC	–	13/02	Integer (Note 3) 1..7 ::= Window size 1..7
Reference	BIS-N-CONNECT SBV_Channel_Open_TC SBV_Channel_Close_TC SBV_Channel_Error_TC	12	–	Not defined in this Recommendation
Req_IB_Called_Address	SBV_Channel_Open_TC	–	08/04	Not defined in this Recommendation
Req_IB_User_Data	SBV_Channel_Open_TC	–	08/07	Not defined in this Recommendation
Req_OB_Called_Address	SBV_Channel_Open_TC	–	08/12	Not defined in this Recommendation
Req_OB_Called_Subaddress	SBV_Channel_Open_TC	–	08/13	Not defined in this Recommendation
Req_OB_User_Data	SBV_Channel_Open_TC	–	08/06	Free format
Reset_Keys	SBV_DFK_TC	–	14/02	Void (9.3.3.3)
Result	BIS-N-CONNECT BIS-N-DISCONNECT SBV_Reset SBV_Channel_Open_TC SBV_Channel_Close_TC SBV_Begin_Application_TC SBV_TPD_Begin_TC	13	–	Integer 0 ::= "Successful" 1 ::= "Not successful" 2 ::= "Application not available" 3 ::= "Application already engaged" 4 ::= "Release not allowed" 5 ::= "TPD not supported" 6 ::= "Illegal Reference parameter"
Reverse_Role_Assignment	BIS-N-CONNECT	14	–	Void
Echo	SBV_Remote_Echo	–	13/03	Integer 0 ::= ON 1 ::= OFF

BCS Basic Coding Structure
ECS Extended Coding Structure

NOTES

1 It is recommended to apply CCITT Recommendation X.121 [7].

2 To permit the distinction of a free-formatted parameter from parameters coded in accordance with 9.2, it is required to set the MSB of the first octet of the free-formatted parameter to 0. If used in conjunction with other parameters which are coded in accordance with 9.2, this parameter shall appear as the last one in the user data field.

3 Other values are for further study.

4 The Error_Detection_Location parameter is local only. Therefore, no value is given in this table. Nevertheless, value 08/02 shall be reserved with respect to future extensions, if any.

9.3.2 Coding of user data fields as defined in other documents

Table 47 shows parameters which are carried in the user data fields of the BIS-N-DATA and BIS-N-Q-DATA services. The coding of these parameters shall be performed according to the various Recommendations as defined in Table 47, without using the coding structure as defined in 9.2.

The X.3 parameter manipulating services employ the parameter X.3_Parameter_List in accordance with CCITT Recommendation X.29 [6].

TABLE 47/T.105

Parameter	BIS-Primitives	Recommendation	Remarks
VTX_Data	BIS-N-DATA	Rec. T.101 [8]	Note 1
TFI_Enq	BIS-N-Q-DATA	Rec. T.101 [8]	Note 2
TFI_Ack	BIS-N-Q-DATA	Rec. T.101 [8]	Note 2
X.3_Parameter_List	BIS-N-Q-DATA	4.4.5/X.29 [6]	

NOTES

1 The VTX_Data parameter is directly mapped on the User data field of a DATA packet and coded in accordance with Recommendation T.101 [8]; it may therefore contain a TFI request (and response) in accordance with Recommendation T.101 [8]. If used in conjunction with the SBV_TPD services, this parameter may also carry processable data without making use of the DDU layer as specified in Recommendation T.101 [8].

2 These parameters are directly mapped onto the Telematic Command Message of the SBV_TFI_TC and coded in accordance with Recommendation T.101 [8]; the support of the particular Videotex presentation data element (VPDE) request is optional.

9.3.3 Service specific coding

9.3.3.1 Coding of the Error_Code parameter

The Error_Code parameter carried in the SBV_TC_Error TC is coded as a sequence of octets; the first octet (coded as an Integer) may take the value:

00/00: “TC not supported”

00/01: “TC erroneous”

00/02: “unknown TC_Event”

00/03: “TC semantic error”

The first case (“TC not supported”) occurs either if the TC value (octet 3) contains a value which is not allowed (see 9.1) or if the value indicates an optional service which is not supported by the receiving side.

NOTE – A given implementation will possibly not distinguish between a non-supported and an invalid TC code point.

In the “TC not supported” case, the following octet shall contain the TC code of the received PDU which is not supported.

EXAMPLE 1: This example (see Figure 9) shows the complete coding of the SBV_TC_Error TC for the case “TC not supported”. It is assumed that the SBV_TPD_Begin service is not implemented.

Octet 1	04/00	Telematic service message
Octet 2	09/14	Syntax-based Videotex
Octet 3	00/01	SBV_TC_Error_TC
Octet 4	00/05	Length of TCM: 5 octets
Octet 5	15/15	Extended coding structure
Octet 6	08/01	Type: Error_Code
Octet 7	00/02	Length: 2 octets
Octet 8	00/00	Value: “TC not supported”
Octet 9	00/08	Value: SBV_TPD_Begin

FIGURE 9/T.105

Example 1 for the coding of the Error_Code parameter

The second case (“TC erroneous”) occurs if the TC is supported, but the coding of one or several of the associated parameters is not correct. This case also occurs if a parameter is mandatory but missing.

In the “TC erroneous” case, the following octets shall contain the TC code of the received TC indication, followed by a list of the erroneous or missing parameter(s). This list shall consist of octets containing the separate type indicator(s) of the erroneous parameter(s) involved. Type indicators which are defined for the Basic Coding Structure (see 9.2.1) shall be coded in the upper four bits of an octet, with the lower four bits set to 0.

EXAMPLE 2: The following example (see Figure 10) shows the complete coding of the SBV_TC_Error TC for the case “TC erroneous”. It is assumed that the minimum set of the SBV_DFK service (i.e. 8 keys) is implemented, but a previously received SBV_DFK indication has tried to manipulate function key 10.

Octet 1	04/00	Telematic service message
Octet 2	09/14	Syntax-based Videotex
Octet 3	00/01	SBV_TC_Error_TC
Octet 4	00/06	Length of TCM: 6 octets
Octet 5	15/15	Extended coding structure
Octet 6	08/01	Type: Error_Code
Octet 7	00/03	Length: 3 octets
Octet 8	00/01	Value: “TC erroneous”
Octet 9	00/10	Value: SBV_DFK
Octet 10	14/01	Value: Function_Keys

FIGURE 10/T.105

Example 2 for the coding of the Error_Code parameter

The third case ("Unknown TC_Event") occurs if within a received PDU either octet 2 or octet 4 or both are not in accordance with this Recommendation.

In the "Unknown TC_Event" case, the following octets shall contain octets 2, 3 and 4 of the received unknown TC_event.

EXAMPLE 3: The following example (see Figure 11) shows the complete coding of the SBV_TC_Error TC for the case "Unknown TC_Event". It is assumed that the recently received octet 2 was not correct.

Octet 1	04/00	Telematic service message
Octet 2	09/14	Syntax-based Videotex
Octet 3	00/01	SBV_TC_Error_TC
Octet 4	00/07	Length of TCM: 7 octets
Octet 5	15/15	Extended coding structure
Octet 6	08/01	Type: Error_Code
Octet 7	00/04	Length: 4 octets
Octet 8	00/02	Value: "Unknown TC_Event"
Octet 9	00/00	Value: Octet 2
Octet 10	00/10	Value: Octet 3
Octet 11	xx/yy	Value: Octet 4

FIGURE 11/T.105

Example 3 for the coding of the Error_Code parameter

The fourth case ("TC semantic error") occurs if a received PDU belongs to a supported service but occurs at the wrong point in time, e.g. a well coded response which belongs to an other service than the one waiting for a response, or a response without a previous request, or a request instead of an expected response.

In the "TC semantic error" case, the following octets shall contain the octet code of the receive PDU in the same manner as in case of "TC not supported".

9.3.3.2 Coding of the DDU_Fall-Back_Mode parameter

The DDU_Fall-Back_Mode parameter is coded as a string.

The possible characters in the string are restricted to:

"A","B","C","D","E","F" and "G" (uppercase only).

The string may contain one or several characters; the characters shall follow the alphabetic order; a character shall not be duplicated.

EXAMPLE: Some possible valid values of the DDU_Fall-Back_Mode parameter:

– "B";

– "ABDEG";

– "ABCDEFG".

9.3.3.3 Coding of the Function_Keys parameter

The SBV_DFK TC Function_Keys parameter is a structured data type. It carries a list of function key definitions. Each function key consists of an identification, an optional user visible name, an optional code sequence and an optional "do-not-forward" indication.

Table 48 defines the coding of the type indicators.

TABLE 48/T.105

Type indicator	Data element
06/00	Function Key
06/01	Identification
06/02	User Visible Name
06/03	Code Sequence
06/04	Do-not-forward

The user visible name may be used by the Terminal Function to inform the user about the purpose of a function key. The code sequence is a sequence of octets which has to be sent to the Access Function when the function key is depressed. By default, the function key shall be associated with a forwarding condition. The presence of a do-not-forward indication defines that the function key shall not be associated with a forwarding condition.

The Terminal Function shall process the conditions in the received order. When setting a specific function key twice in one Telematic Command, the value of the last received definition shall be used. Reset commands which are sent in between function key lists shall be processed.

The list of function keys is coded as a structure. Each element of this structure is coded in a TLV (type-length-value) form, which conforms to the extended coding structure as defined in 9.2.2.

The identification is coded as an integer. For terminals supporting the SBV_DFK service, at least values 1 to 8 shall be supported. The user visible name shall be encoded as a string of the indicated length. The code sequence shall be encoded as a sequence of octets of the indicated length. The values of these octets are not limited to a specific data syntax. The do-not-forward indication shall be encoded as a void type.

The following formal specification gives the syntax of the Function_Keys data-structure:

```
Function_Keys        ::=   "15/15" "14/01"     Length Function_Key_List

Function_Key_List    ::=   Function_Key        Function_Key_List
                           |   /* empty */

Function_Key         ::=   "06/00"             Length Identification
                                                         User_visible_name
                                                         Code_sequence
                                                         Do_not_forward

Identification       ::=   "06/01"             Length Integer

User_visible_name    ::=   "06/02"             Length String
                     |   /* empty */

Code_sequence        ::=   "06/03"             Length Octet_sequence
                     |   /* empty */

Do_not_forward       ::=   "06/04" "00/00"
                     |   /* empty */

Length               ::=   "Octet conforming to the length type in 9.2.2"

Integer              ::=   "Integer conforming to 9.2.3.1"

String               ::=   "String conforming to 9.2.3.2"

Octet_sequence       ::=   "Octet sequence conforming to 9.2.3.4"
```

The following examples illustrate the use of the SBV_DFK service, showing the complete coding for the SBV_DFK TC:

EXAMPLE 1: After deleting all current definitions, associate key 1 with the user visible name “F1” and the code string “Code1”, and key 10 with the user visible name “F10” and the code sequence “Code2” (see Figure 12).

04/00	Telematic service message		
09/14	Syntax-based Videotex		
00/10	SBV_DFK_TC		
02/07	Length of TCM: 39 octets		
	15/15	Extended coding structure	
	14/02	Reset	
	00/00	length = 0	
	15/15	Extended coding structure	
	14/01	function key list	
	02/01	length = 33	
	06/00	function key	
	00/14	length = 14	
		06/01	identification
		00/01	length = 1
		00/01	value = 1
		06/02	user visible name
		00/02	length = 2
		“F”	value = “F1”
		“1”	
		06/03	code sequence
		00/05	length = 5
		“C”	value = “Code1”
		“o”	
		“d”	
		“e”	
		“1”	
	06/00	function key	
	00/15	length = 15	
		06/01	identification
		00/01	length = 1
		00/10	value = 10
		06/02	user visible name
		00/03	length = 3
		“F”	value = “F10”
		“1”	
		“0”	
		06/03	code sequence
		00/05	length = 5
		“C”	value = “Code2”
		“o”	
		“d”	
		“e”	
		“2”	

FIGURE 12/T.105

Example 1 for the coding of the SBV_DFK TC PDU

EXAMPLE 2: Erase reset key 10 and overwrite key 1 with “FN” for the user visible name and code string “New”, which is not a forwarding condition (see Figure 13).

04/00	Telematic service message			
09/14	Syntax-based Videotex			
00/10	SBV_DFK_TC			
01/08	Length of TCM: 24 octets			
	15/15 14/01 01/05	Extended coding structure function key list length = 21		
		06/00 00/03	function key length = 14	

			06/01
			00/01
			00/10
				identification
				length = 1
				value = 10

		06/00 00/14	function key length = 14	

			06/01
			00/01
			00/01
			06/02
			00/02
			“F”
			“N”
				identification
				length = 1
				value = 1
				user visible name
				length = 2
				value = “FN”

			06/03
			00/03
			“N”
			“e”
			“w”
				code sequence
				length = 3
				value = “New”

			06/04 00/00	Do not_forward length = 0

FIGURE 13/T.105

Example 2 for the coding of the SBV_DFK TC PDU

10 Use of CCITT Recommendation X.3 parameters

CCITT Recommendation X.3 [3] parameters shall be used as described in the relevant CCITT Recommendations. Additional rules are given in this Recommendation. The values of the X.3 parameters given in this Recommendation shall be changed only through the use of the SBV_Set_Param, SBV_Set_Read_Param, and SBV_Reset services. The values shall not be changed by the Terminal Function itself.

10.1 Selectable functions provided by Syntax-based Videotex Terminals

For a global description of these functions, refer to the Terminal Model described in 6.2.

10.2 List of X.3 parameters

For the description of the X.3 parameters refer to CCITT Recommendation X.3 [3], 1.4. The following parameters are selected (see Table 49) for the Syntax-based Videotex Terminal Function.

TABLE 49/T.105

Supported CCITT Recommendation X.3 [3] parameters

Parameter reference number	Parameter description
2 (M)	Echo
3 (M)	Selection of data forwarding characters (Note 1)
4 (M)	Selection of idle timer delay (Note 2)
11 (M)	Binary speed (Note 3)
23 (O)	Size of input field
24 (O)	End of frame
25 (O)	Selection of extended data forwarding signals
26 (O)	Display interrupt
28 (O)	Diacritic character editing
29 (O)	Extended echo mask

M Mandatory
O Optional

NOTES

1 In addition to CCITT Recommendation X.3 [3], the values 1 and 16 are mandatory values in this Recommendation.

2 In addition to CCITT Recommendation X.3 [3], the parameter value 1 is mandatory in this Recommendation. Some Syntax-based Videotex Terminals may interpret this value in the sense of “as soon as possible”.

3 This Parameter shall be used to indicate the binary speed between the TF and the AF. This is based on parameter reference 11 as defined in CCITT Recommendation X.3 [3]. Possible values can be given by a Recommendation related to the lower layers.

11 Bearer Independent Service (BIS) for Syntax-based Videotex

11.1 Introduction

The upper layers In-Band protocol of Syntax-based Videotex (higher than 3) makes use of CCITT Recommendation X.29 [6] based procedures. In some cases, additional procedures, e.g. to establish a physical link or e.g. to handle the access to the network are used, named Out-Band procedures.

The purpose of this clause is to define a Bearer Independent Service for the application of Syntax-based Videotex, named BIS.

The In-Band procedures used to support the BIS are defined by this clause.

The Out-Band procedures depend strongly on the network type, the method to establish a physical link or to access the network. They cannot be given entirely here. Subclause 11.3.1 provides for an anchorage to be used by other documents being specific to a certain network type.

NOTE – All physical layer aspects are assumed to be given by those documents as well.

11.2 Scope and field of application

The BIS is defined for the specific application of Syntax-based Videotex as service user, but other applications are not excluded.

The principal objectives of the BIS definition are:

a) to specify the characteristics of a conceptual Network Service, hiding from the Network Service user how the service is provided;

b) to realize the service through the use of X.25/PLP, and some Out-Band protocol if appropriate.

The BIS is defined in terms of:

1) the service primitives and their associated parameters;

2) X.25 packets and fields;

3) the interrelationship between, and the valid sequences of, these primitives and packets.

NOTE 1 – The use of Out-Band specific protocol elements is assumed to be determined by documents related to the Out-Band procedures.

Apart from the In-Band/Out-Band coordination and the qualified data transfer service (data with Q-bit = 1), the functions and procedures defined in this clause are a subset of the OSI Connection-mode Network Service. Quality of service functions, although out of the scope of this Bearer Independent Service, are not precluded.

For the In-Band protocol, ISO/IEC 8208 [15] or CCITT Recommendation X.25 (1980) [4] or later versions of Recommendation X.25 shall be used with the following rules:

NOTE 2 – The option of using some facilities may only be employed if the version is either ISO/IEC 8208 [15] or CCITT Recommendation X.25 (1984) [5] or later.

– Only the X.25 Virtual Call service applies to the BIS. The use of X.25 Permanent Virtual Circuit service for the purposes of the BIS is excluded.

– The BIS is provided within a DTE (in X.25 terminology), by the use of X.25 procedures across a DTE/DCE or a DTE/DTE interface as well.

NOTE 3 – The physical layer aspects for the In-Band procedures will depend strongly on the network type. They are assumed to be determined by documents related to the Out-Band procedures.

In particular, the BIS provides for the following:

a) *Independence of underlying transmission media*

The BIS relieves BIS users of all concerns regarding how various sub-networks are used to provide the BIS or the access to them. The BIS hides from the BIS user differences in the transfer of data over heterogeneous sub-networks.

b) *End-to-end transfer*

The BIS provides for transfer of BIS user data between BIS users in **end systems**. All routing and relaying functions are performed by the BIS provider.

c) *Transparency of transferred information*

The BIS provides for the transparent transfer of octet-aligned BIS user data. It does not restrict the contents, format or coding of the information, nor does it ever need to interpret its structure or meaning.

The BIS offers the following features to an BIS user:

1) The means to establish a Network Connection (NC) with another BIS user for the purpose of transferring BIS user data by means of data transfer primitives. More than one NC may exist between the same pair of BIS users.

2) The means of transferring BIS user data in so-called BIS Data Units (BIS-DUs) in sequence on a NC. The transfer of BIS-DUs, which consist of an integer number of octets, is transparent, in that the boundaries of BIS-DUs and the contents of BIS-DUs are preserved unchanged by the BIS, and there are no constraints on the BIS-DU content imposed by the BIS. Two different types of user data transfer services are provided:

   – the “unqualified” data transfer service;

   – the “qualified” data transfer service.

3) The means of transferring expedited data in so-called BIS-Interrupt BIS-DUs, which consist of an integer number of octets, is transparent, in that the boundaries of BIS-DUs and the contents of BIS_DUs are preserved unchanged by this BIS, and there are no constraints on the BIS-DU content imposed by the BIS.

4) The means by which the NC can be returned to a definite state and activities of the two BIS users synchronized by use of a reset service.

5) The unconditional, and therefore possibly destructive, release of an NC by either of the BIS users or by the BIS provider.

The BIS does not specify individual implementations or products nor does it constrain the implementation of entities and interfaces within a computer system. In other words, BIS defines an Abstract Service interface; the internal representation of its Abstract Service Primitives is a local matter.

11.3 General operation of the BIS

11.3.1 Out-Band Procedures for Environment-dependent operation

The X.25/PLP is used to provide the BIS in an end system connected directly to an X.25 packet-switched network (dedicated access), or it can be used in environments, where an intermediate access path has to be provided either between the endsystem and the X.25 packet-switched network or between the two end systems (e.g. through an ISDN).

The establishment of a physical link or an intermediate access path, if necessary, is performed by a sub-function of the NL normally called coordination function. The description of the operation of the coordination function depends on the environment and is outside the scope of this Recommendation.

It is assumed however, that, before the procedures of the BIS defined in the subclauses below apply, the following steps are taken, when a BIS-N-CONNECT request is received by the NL entity:

a) The physical access path shall be established by the NL entity if necessary. Possible retries after a first failure are a local matter. After successful establishment, the initialization defined in b) and c) below, shall be performed.

b) A Data Link Connection shall be initiated by the NL entity if it does not already exist.

c) The X.25/PLP RESTART procedure shall be initiated by the NL entity, if the Data Link Connection was established in step b).

It is not within the scope of this Recommendation to indicate how the NL entity is informed of the outcome of the procedures a), b) and c), but it is assumed that the NL entity is informed of the outcome. If the outcome is positive, the procedures of the BIS defined in the subclauses below apply. If the outcome is negative, a BIS-N-DISCONNECT indication is issued to the BIS user.

It is not within the scope of this Recommendation whether an existing access path is used for the establishment of a new Virtual Call supporting a BIS-N-CONNECT request (using, e.g. more than one LC), or whether a new access path has to be established.

It is assumed that the SBV_Establish address parameters the names of which start with “OB_” form part of the BISAP address and are used to establish the access path. Furthermore, it is as assumed at the contents of the other SBV_Establish parameters the names of which start with “OB_” are conveyed by some Out-Band data transfer means (e.g. UUS in case of ISDN), if available, before the In-Band procedures take place.

In a similar way as above for the BIS-N-CONNECT request primitive, it is assumed, that the initialization procedures in a), b) and c) have been completed before an BIS-N-CONNECT indication can be signalled to the BIS user.

It is further assumed, that in case the environment in which the X.25/PLP operates prematurely breaks down (i.e. while one or more NCs are established or in the process of being established), then the NL entity signals for each established NC and for each NC in the process of being established, a BIS-N-DISCONNECT indication to the BIS user.

The particular values of the Originator and Reason parameters passed to the BIS user in case of an unsuccessful path establishment or a premature breakdown of the access path are outside the scope of this Recommendation. However, the following rules shall be obeyed:

1) If an access network connection or a VC cannot be established for local reasons (e.g. no B-channel available if the access network is an ISDN, or if no LC is available), the "originator" value shall be "local" and no "reason" value is specified.

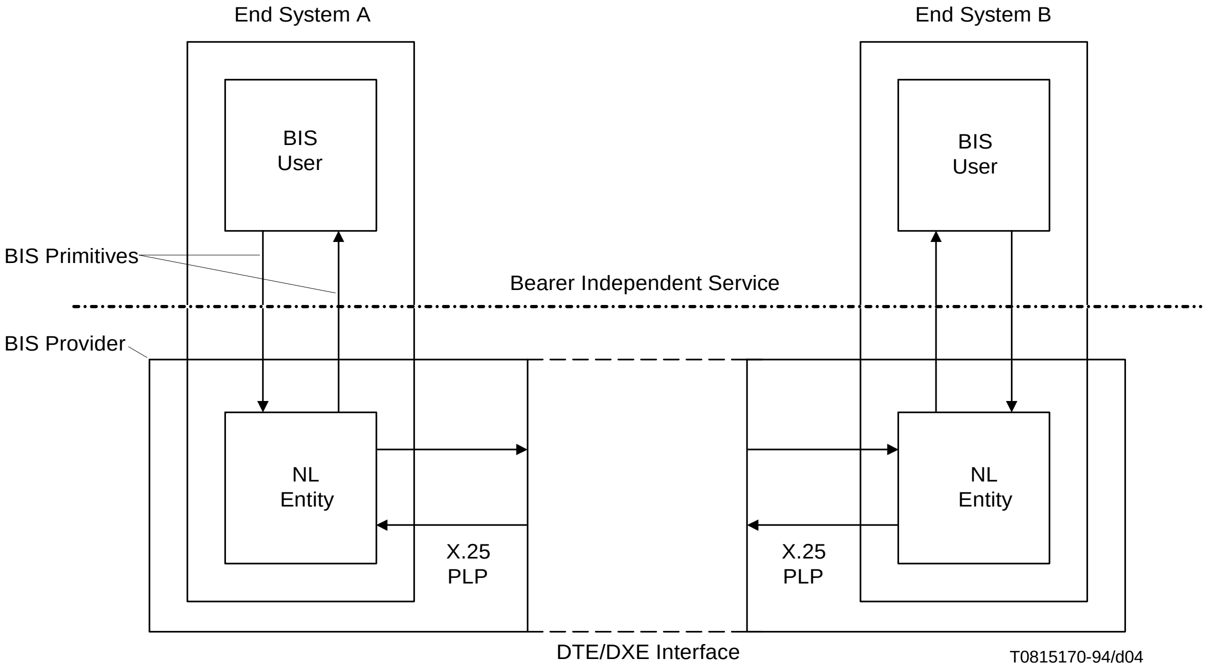
2) If the access network connection establishment cannot be completed, or there is a premature breakdown of the access path, the "originator" value shall be according to the access network the end system is connected to (e.g. "ISDN" if the access network is an ISDN). The "reason" value depends on the access network.

11.3.2 Translation of primitives and parameters of In-Band Procedures

As shown in Figure 14, the BIS provider (more particularly, the NL entity in the end system) must provide a translation between:

a) the primitives and parameters of the BIS; and

b) the packets and associated fields of the X.25/PLP.



NOTE – Out-Band related matters are not shown here.

FIGURE 14/T.105

Operation of the BIS

Request and response primitives are translated into packets (where appropriate) to be transmitted across the DTE/DXE interface by the NL entity. Received packets, where appropriate are translated by the NL entity into indication and confirm primitives.

There is also a relationship between some local mechanism used to identify a particular NC and a LC number used to identify a particular virtual circuit. This relationship is a local matter and not subject to standardization.

NOTE – There may be X.25/PLP compatible functions implemented in an end system which are outside the scope of this Recommendation, but which may have an effect on the provision of the BIS, e.g. if an end system chooses to consider QOS aspects and a certain QOS cannot be achieved for a particular VC, a BIS-N-CONNECT request may be refused.

11.4 Service primitives and parameters

The following subclauses define the service primitives applicable to the various phases occurring during the provision of the BIS on a particular NC, and the parameters associated to them.

The mapping given in this subclause shall be used when the mapping is In-Band caused. The Out-Band caused mapping is not defined by this Recommendation.

NOTE – It is assumed that the Out-Band caused mapping is covered by documents being specific to a certain network (see 11.3.1).

Sometimes, the availability of a parameter for a service primitive depends on the support of the Fast Select facility for the NC. The knowledge whether Fast Select is supported, is a local matter and not indicated by a parameter of the BIS.

11.4.1 NC establishment primitives and parameters

NC establishment may have Out-Band related aspects (see 11.3.1) which are not defined here. The rules defined in this subclause shall be respected for all In-Band related aspects of the NC establishment.

11.4.1.1 Association of primitives and parameters

Table 50 shows the service primitives applicable in the NC establishment phase and the parameters associated to them.

TABLE 50/T.105

NC Establisment primitives and parameters

Parameters	BIS-N-CONNECT request	BIS-N-CONNECT indication	BIS-N-CONNECT response	BIS-N-CONNECT confirm
Called Address	X	X (=)		
Calling Address	X	X (=)		
Responding Address			X	X (=)
BIS-user-data	X	X (=)	X	X (C=)

(=) The parameter value supplied in an indication or confirm primitive is identical to the value supplied in the corresponding request or response primitive occurring at the peer BISAP.

(C) The parameter is not present on the primitive for every NC; the parameter definition describes the conditions under which the parameter is present or absent.

11.4.1.2 Parameter definitions

- **Called Address** – The Called Address parameter contains one or more addresses, used to identify the BISAP to which the NC is to be established.
- **Calling Address** – The Calling Address parameter contains one or more addresses, used to identify the BISAP from which the NC is requested.
- **Responding Address** – The Responding Address parameter contains one or more addresses, used to identify the BISAP to which the NC has been established.

NOTES

1 Addresses may be implicitly associated with the BISAP at which the primitive is used.

2 Depending on the environment (see 11.3.1) more than one address may be necessary to identify a BISAP. As a consequence, the abstract Called Address parameter may comprise more than one physical part, mapped onto more than one uderlying protocol element field

3 Mechanisms operating within the BIS provider, e.g. call redirection, may result in address parameters in corresponding primitives not being identical in the following cases:

a) the responding address parameter in the BIS-N-CONNECT response may not necessarily be the same as the called address parameter in the BIS-N-CONNECT indication;

b) the responding address parameter in the BIS-N-CONNECT confirm may not necessarily be the same as the called address parameter in the BIS-N-CONNECT request.

– **BIS-user-data** – The BIS-user-data parameter is used to transfer BIS user data (an integer number of octets) between BIS users, without modification by the BIS provider.

NOTE – The BIS-user-data abstract parameter may comprise more than one physical part, mapped onto more than one underlying protocol element field, depending on the environment. For instance, in case of an ISDN, some part may be mapped onto the UUI part of a SETUP message, an other part may be mapped onto the Call User Data field of an X.25 CALL REQUEST. The BIS-user-data abstract parameter is not necessarily a continuous block of data.

The BIS-user-data parameter information that makes use of the X.25/PLP Fast Select facility (see 11.5.3) is subject to the following rule: for a BIS-N-CONNECT request or indication, the allowed parameter length is between 0 and 128 octets when Fast Select is used, otherwise the allowed length is between 0 and 16 octets.

The BIS-user-data parameter is optional for the BIS-N-CONNECT response and confirm; in these cases it may only be used, if the Fast Select Acceptance facility is supported and the Fast Select facility is used for the NC.

11.4.2 NC release primitives and parameters

NC release may have Out-Band related aspects (see 11.3.1) which are not defined here. The rules defined in this subclause shall be respected for all In-Band related aspects of the NC release.

11.4.2.1 Association of primitives and parameters

Table 51 shows the service primitives applicable in the NC release phase and the parameters associated to them.

TABLE 51/T.105

NC Release primitives and parameters

Parameters	BIS-N-DISCONNECT request	BIS-N-DISCONNECT indication
Originator	X	X
Reason	X	X
Responding Address	X (C)	X (C=)
BIS-user-data	X (C)	X (C=)

(=) The parameter value supplied in an indication primitive is identical to the value supplied in the corresponding request primitive occurring at the peer BISAP.

(C) The parameter is not present on the primitive for every NC; the parameter definition describes the conditions under which the parameter is present or absent.

11.4.2.2 Parameter definitions

- **Originator** – The Originator parameter indicates the source of the NC release. Its value shall be an X.25 cause.

  NOTE – For not In-Band caused NC releases, the Originator parameter may also take other values like "local" or Out-Band related values like "PSTN" or "ISDN".

- **Reason** – The Reason parameter gives information about the cause of the NC release. The value conveyed in this parameter shall be an X.25 diagnostic.

  NOTES

  1 For not In-Band caused NC releases, the Reason parameter may also take other values like "unspecified" or Out-Band related values like a Q.931 cause in case of an ISDN.

  2 Further information on the value to be selected for the originator and reason parameters in case of an In-Band caused NC release can be found in 11.7 "Network connection release phase".

- **Responding Address** – The Responding Address parameter is an optional parameter which may only be used, if the Fast Select facility is supported for the NC. Furthermore, it is only present in case the primitive is used to indicate rejection of an NC establishment attempt by a BIS user. In this case it contains one or more addresses, used to identify the BISAP from which the BIS-N-DISCONNECT request was issued.

  NOTES

  1 Addresses may be implicitly associated with the BISAP at which the primitive is used.

  2 Depending on sub-networks between BIS users, e.g. PSTNs or ISDNs, more than one address may be necessary to identify a BISAP. The contents of the address parameter is outside the scope of this Recommendation.

- **BIS-user-data** – The BIS-user-data parameter is used to transfer BIS user data (an integer number of octets) between BIS users, without modification by the BIS provider. The parameter is optional and may only be used, if the Fast Select facility is used for the NC.

  If the Fast Select facility is used for the NC, the maximum parameter length is 128 octets for each primitive.

  The BIS-user-data sent is lost if NC release is simultaneously invoked by either the BIS provider or the intended receiving BIS user.

11.4.3 Unqualified data transfer primitives and parameters

Unqualified data transfer has no Out-Band related aspects.

11.4.3.1 Association of primitives and parameters

Table 52 shows the service primitives applicable to the transfer of unqualified data in the data transfer phase, and the parameters associated to them.

TABLE 52/T.105

Data Transfer (unqualified) primitives and parameters

Parameters	BIS-N-DATA request	BIS-N-DATA indication
BIS-user-data	X	X (=)
(=) The parameter value supplied in an indication primitive is identical to the value supplied in the corresponding request primitive occurring at the peer BISAP.		

11.4.3.2 Parameter definitions

- **BIS-user-data** – The BIS-user-data parameter is used to transfer BIS user data (one or more octets) between BIS users, without modification by the BIS provider.

11.4.4 Qualified data transfer primitives and parameters

Qualified data transfer has no Out-Band related aspects.

11.4.4.1 Association of primitives and parameters

Table 53 shows the service primitives applicable to the transfer of qualified data in the data transfer phase, and the parameters associated to them.

TABLE 53/T.105

Qualified data transfer primitives and parameters

Parameters	BIS-N-Q-DATA request	BIS-N-Q-DATA indication
BIS-user-data	X	X (=)
(=) The parameter value supplied in an indication primitive is identical to the value supplied in the corresponding request primitive occurring at the peer BISAP.		

11.4.4.2 Parameter definitions

- **BIS-user-data** – The BIS-user-data parameter is used to transfer qualified BIS user data (one or more octets) between BIS users, without modification by the BIS provider.

11.4.5 Interrupt data transfer primitives and parameters

Interrupt data transfer has no Out-Band related aspects.

11.4.5.1 Association of primitives and parameters

Table 54 shows the service primitives applicable to the transfer of interrupt data in the data transfer phase, and the parameters associated to them.

TABLE 54/T.105

Interrupt data transfer primitives and parameters

Parameters	BIS-N-INTERRUPT request	BIS-N-INTERRUPT ndication
BIS-interrupt-user-data	X	X (=)
(=) The parameter value supplied in an indication primitive is identical to the value supplied in the corresponding request primitive occurring at the peer BISAP.		

11.4.5.2 Parameter definitions

- **BIS-interrupt-user-data** – the BIS-interrupt-user-data parameter is used to transfer interrupt BIS user data (one or more octets) between BIS users, without modification by the BIS provider.

11.4.6 Reset primitives and parameters

BIS reset has no Out-Band related aspects.

11.4.6.1 Association of primitives and parameters

Table 55 shows the service primitives applicable to the reset service in the data transfer phase, and the parameters associated to them.

TABLE 55/T.105

Reset primitives and parameters

Parameters	BIS-N-RESET request	BIS-N-RESET indication	BIS-N-RESET response	BIS-N-RESET confirm
Originator	X	X	–	–
Reason	X	X	–	–

11.4.6.2 Parameter definitions

- **Originator –** The Originator parameter indicates the source of the NC reset. Its value is directly mapped to/from the X.25 cause of the related RESET REQUEST/INDICATION packet.
- **Reason** – The Reason parameter gives information indicating the cause of the NC reset. Its value is directly mapped to/from the X.25 diagnostics of the related RESET REQUEST/INDICATION packet.

  Further information on the values to be selected for the originator and reason parameter can be found in 11.10.2.3, “Originator and Reason”.

11.4.7 Relation of primitives at the two NC endpoints

A primitive issued at one NC endpoint will, in general, have consequences at the other NC endpoint. The relations of primitives of each type to primitives at the other NC endpoint are summarized in Figure 15.

11.4.8 Sequence of primitives at one NC endpoint

The valid overall sequences of primitives at an NC endpoint are defined in the state transition diagrams of Figure 16. In the diagrams:

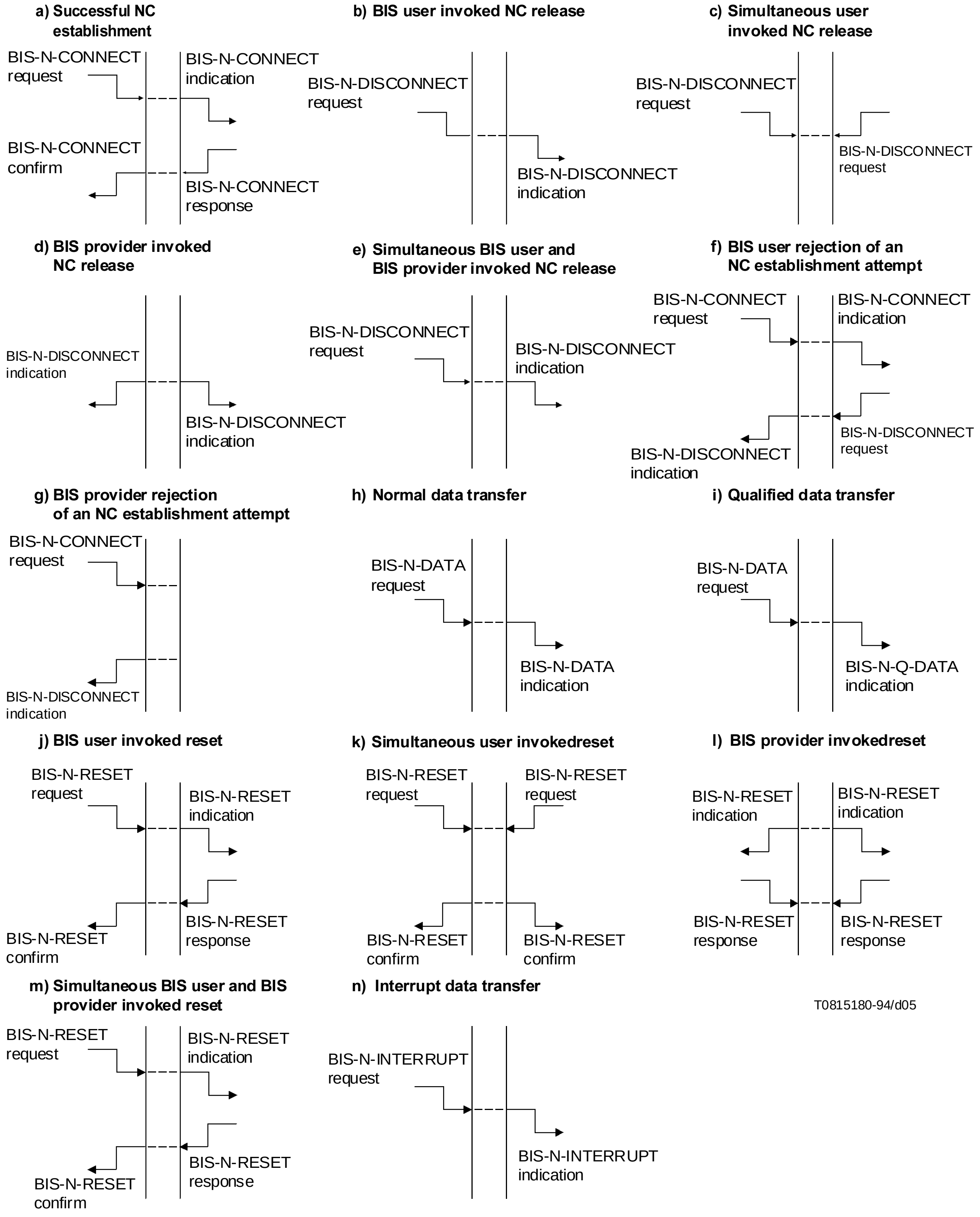
a) A primitive which is not shown as resulting in a transition (from one state to the same state or from one state to a different state), is not permitted in that state. The actions to be performed in this case are a local matter and not subject to standardization.

b) The labelling of the states “BIS user invoked reset pending” (state 5) and “BIS provider invoked reset pending” (state 6) indicates the party which started the local interaction, and does not necessarily reflect the value of the “originator” parameter in the associated BIS-N-RESET primitive.

c) The Idle (state 1) reflects the absence of an NC. It is the initial and final state of any sequence, and once it has been re-entered, the NC is released.

d) The use of a state transition diagram to describe the allowable sequences of service primitives does not impose any requirements or constraints on the internal organization of any implementation of the BIS.

Figure 16 shows for each state the permitted service primitives to be issued, and the state to be entered afterwards.



NOTE – See 11.10.2.2 for the relation between BIS-N-RESET confirm and response

FIGURE 15/T.105

Summary of BIS service primitive time sequence diagrams

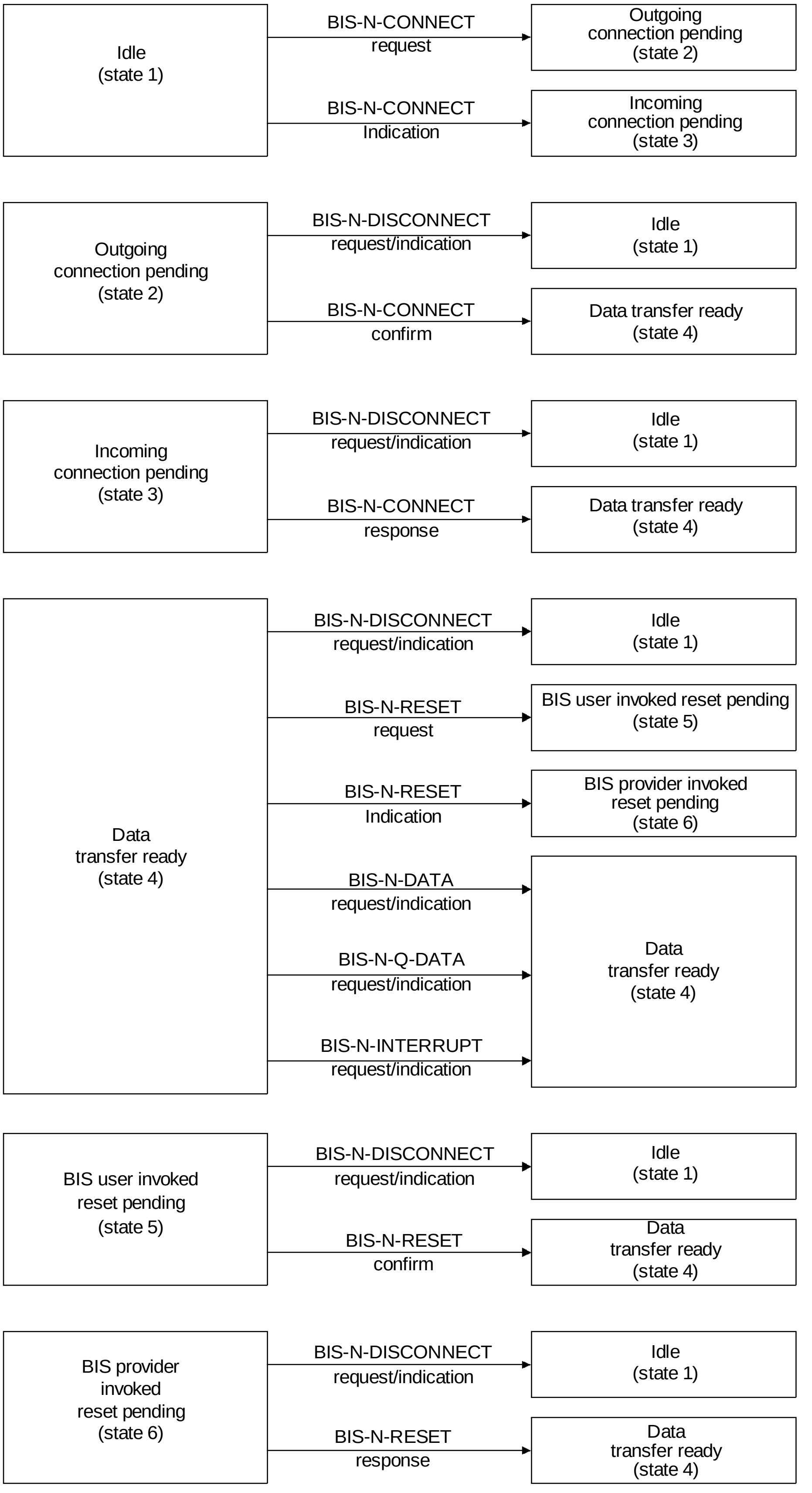


FIGURE 16/T.105

state transition diagram

11.5 Elements of the X.25/PLP used to support the BIS

The X.25/PLP, as defined by ISO/IEC 8208 [15], provides the realization for the transparent transfer of qualified and unqualified data between BIS users. The elements of this protocol to be considered here are:

a) virtual circuits;

b) the packet types and fields to be mapped to the primitives and parameters of the BIS;

c) optional user facilities and CCITT-specified DTE facilities.

No facilities are mandatory for the operation of the BIS, some facilities are indicated as an option, and no facility except "Fast Select with restriction on response" is excluded.

11.5.1 Types of virtual circuits

Of the two types of virtual circuits defined in ISO/IEC 8208 [15], only the use of Virtual Calls is mapped to the NC establishment and release phases of the BIS (see 11.6 and 11.7).

NOTE – A corresponding mapping for permanent virtual circuits is not defined.

11.5.2 Packets and fields of the X.25/PLP used to support the BIS

Table 56 below lists the X.25/PLP packets and associated fields that shall be used when supporting the BIS.

TABLE 56/T.105

X.25 packets and fields to support the BIS

Packets (Note 1)	Fields (Note 2)
CALL REQUEST INCOMING CALL CALL ACCEPTED CALL CONNECTED	Address field Facility field Calling/Called User Data field (Note 3)
CLEAR REQUEST CLEAR INDICATION	Clearing Cause field Diagnostic Code field Address field Facility field (Note 4) Clear User Data field (Note 3)
DATA	Q-bit M-bit User Data field (Note 3)
INTERRUPT RESET REQUEST RESET INDICATION	Interrupt User Data field (Note 3) Resetting Cause field Diagnostic Code field
RESTART REQUEST RESTART INDICATION	Restarting Cause field Diagnostic Code field

NOTES

1 The packets shown in the table are used to support the primitives of the BIS. Other packets not contained in the table may be essential, but have no relation to the provision of the BIS.

2 The information in the fields shown in the table has a direct relationship to the parameters associated with the primitives of the BIS. Other fields not shown in the table (e.g. Logical Channel Identifier, Packet Type Identifier) are essential to the use of the appropriate packets.

3 All user data fields are octet aligned.

4 Facilities are optional in the BIS. See also 11.5.3 below.

11.5.3 Facilities of the X.25/PLP used to support the BIS

No facility is required to support the BIS, however the following facilities have a relation to the provision of the BIS:

- Fast Select, no restriction on response;
- Fast Select Acceptance;
- Calling/Called Address Extension.

The usage of **Fast Select with no restriction on response** is optional. Two coding procedures for Call User Data to be transferred in call set-up packets are defined at the higher layers, dependent on whether Fast Select is used (on a per call basis) or not. The usage is a local matter. If it is not used, the Clear User Data Field is empty and the corresponding parameter(s) cannot be used for information exchange between the higher layers and the Network Layer (see Notes in 7.2.1, 7.2.2, 8.3.1 and 8.3.2).

The **Fast Select with restriction on response** facility shall not be used.

The use of the **Calling/Called Address Extension** facility to identify a BISAP unambiguously depends on local operation requirements and is optional.

11.6 Network connection establishment phase

NOTE – Out-Band related matters are out of the scope of this Recommendation (see 11.3.1).

11.6.1 Primitive/parameter and packet/field relationships

Table 57 shows the relationships between the primitives parameters and the packets/fields for the network connection establishment phase.

TABLE 57/T.105

BIS/X.25 mapping for the network connection establishment phase

BIS	X.25/PLP
PRIMITIVES	PACKETS
BIS-N-CONNECT request BIS-N-CONNECT indication BIS-N-CONNECT response BIS-N-CONNECT confirm	CALL REQUEST INCOMING CALL CALL ACCEPTED CALL CONNECTED
PARAMETERS	FIELDS
Called Address	Called DTE Address field Called Address Extension facility (Note 1)
Calling Address	Calling DTE Address field Calling Address Extension facility (Note 1)
Responding Address	Called DTE Address field Called Address Extension facility (Note 1)
BIS-user-data	Call and Called User Data field Fast Select facility (Note 2) Address field (Note 3)

NOTES

1 The use of the Calling/Called Address Extension facility is not mandatory for a given NC and depends on local operation requirements.

2 The support of the Fast Select facility or its use for a given NC is not mandatory. The selection of the use or non-use is a local matter.

3 The information which is related to the SBV protocol parameter IB_Called_Address may occur only when In-Band Addressing is not used.

11.6.2 Procedures

11.6.2.1 General Function

The NC establishment service primitives can be used to establish an NC, provided the BIS users exist and are known to the BIS provider.

Simultaneous BIS-N-CONNECT primitives at the two BISAPs are handled independently by the BIS provider. They may result in two, one or zero NCs. Part a) in Figure 15 indicates a valid sequence of BIS primitives for successful NC establishment.

11.6.2.2 Primitive/packet mapping

When an NL entity receives a BIS-N-CONNECT request from a BIS user and is able to set up a VC, it transmits a CALL REQUEST packet across the DTE/DXE interface. Otherwise, it issues a BIS-N-DISCONNECT indication to the BIS user.

NOTE 1 – A reason for not being able to set up a VC could be that currently no LC is available.

When an NL entity receives a BIS-N-CONNECT response from a BIS user, it transmits a CALL ACCEPTED packet across the DTE/DXE interface.

When an NL entity receives an INCOMING CALL packet and is able to accept the call, it signals a BIS-N-CONNECT indication to the BIS user. The NL entity may also for internal reasons reject the call by sending a CLEAR REQUEST packet across the DTE/DXE interface. In this case no service primitive is passed to the BIS user.

NOTE 2 – A possible reason for rejecting the call could be that the NL entity chooses to consider QOS aspects and the required QOS cannot be achieved for this call.

When an NL entity receives a CALL CONNECTED packet, it signals a BIS-N-CONNECT confirm primitive to the BIS user.

11.6.2.3 Addresses

The contents of the address fields are determined by local operation.

11.6.2.4 User data

The call user data field of X.25/PLP CALL REQUEST and INCOMING CALL packets is used to transfer the BIS-user-data of BIS-N-CONNECT request and indication primitives, respectively. The called user data field of X.25/PLP CALL ACCEPTED and CALL CONNECTED packets is used to transfer the BIS-user-data of BIS-N-CONNECT response and confirm primitives, respectively.

NOTE – In CALL ACCEPTED and CALL CONNECTED packets, the Called User Data field only exists if the Fast Select facility was used in the corresponding CALL REQUEST or INCOMING CALL packet, which is an option for the BIS.

11.7 Network connection release phase

NOTE – Out-Band related matters are out of the scope of this Recommendation (see 11.3.1).

11.7.1 Primitive/parameter and packet/field relationships

Table 58 shows the relationships between the primitives/parameters and the packets/fields for the network connection release phase.

TABLE 58/T.105

BIS/X.25 mapping for the network connection release phase

BIS	X.25/PLP
PRIMITIVES BIS-N-DISCONNECT request BIS-N-DISCONNECT indication	PACKETS CLEAR REQUEST CLEAR INDICATION RESTART INDICATION CLEAR REQUEST RESTART REQUEST
PARAMETERS	FIELDS
Originator and Reason	Cause and Diagnostic Code fields
Responding Address	Called DTE Address field Called Address Extension facility (Note 1)
BIS-user-data	Clear User Data field Fast Select facility (Note 2)

NOTES

1 The use of the Called Address Extension facility depends on local operation requirements.

2 The use of the facility is not mandatory. The selection of the use or non-use is a local matter. However, it has an effect on the operation of the higher layers.

11.7.2 Procedures

11.7.2.1 General Function

The NC release service primitives are used to an NC. The NC release may be performed:

a) by either or both of the BIS users to release an established NC;

b) by the BIS provider to release an established NC; all failures to maintain an NC are indicated this way;

c) by the called BIS user to reject a BIS-N-CONNECT indication;

d) by the BIS provider to indicate its inability to establish a requested NC.

Valid sequences of service primitives including NC release service primitives are shown in parts b) to g) of Figure 15.

NC release is permitted at any time regardless of the current phase of the NC. Once an NC release procedure has been invoked, the NC will be released: a request for NC release cannot be rejected. After NC release has been invoked at one NC endpoint, the BIS provider may discard any unqualified and qualified BIS user data that has not yet been delivered at the other NC endpoint and may cause any uncompleted sequence of primitives for NC establishment or reset to remain uncompleted.

11.7.2.2 Primitive/packet mapping

When an NL entity receives a BIS-N-DISCONNECT request primitive from a BIS user related to an NC established or in the process of being established (states 2 to 6), it transmits a CLEAR REQUEST packet across the DTE/DXE interface.

If an NL entity detects an error in the operation of the X.25/PLP for which its action is to clear the VC assigned to an NC, then it transmits a CLEAR REQUEST packet across the DTE/DXE interface and passes a BIS-N-DISCONNECT indication primitive to the BIS user.

If an NL entity detects an error in the operation of the X.25/PLP for which its action is chosen to restart all VCs, then it transmits a RESTART REQUEST packet across the DTE/DXE interface and passes a BIS-N-DISCONNECT indication primitive for each NC established or in the process of being established to the BIS user.

When an NL entity receives a CLEAR REQUEST or CLEAR INDICATION packet for a VC assigned to an NC established or in the process of being established, it signals a BIS-N-DISCONNECT indication primitive to the BIS user and also transmits a CLEAR CONFIRMATION packet across the DTE/DXE interface.

If the received CLEAR REQUEST or CLEAR INDICATION packet is the answer to a previously transmitted CALL REQUEST packet, the NL entity may alternatively retry the call before sending a BIS-N-DISCONNECT to the BIS user. The number of retries and the time interval between retries are a local matter. If multiple attempts to establish an NC are all unsuccessful, a BIS-N-DISCONNECT indication primitive is issued to the BIS user.

NOTE – The NL entity may examine the clearing cause to decide whether a retry of the call is attempted or not.

When an NL entity receives a RESTART REQUEST or RESTART INDICATION packet, it signals a BIS-N-DISCONNECT indication primitive for each NC established or in the process of being established to the BIS user, and also transmits a RESTART CONFIRMATION packet across the DTE/DXE interface.

11.7.2.3 Originator and Reason

In case of a BIS-N-DISCONNECT request, the cause code to be inserted into the CLEAR REQUEST packet shall be “DTE originated”. The “reason” parameter shall contain any diagnostic code compatible to the cause, and the diagnostic code field of the CLEAR REQUEST packet shall be coded according to the “reason” parameter.

In case of a BIS-N-DISCONNECT indication after receipt of a CLEAR REQUEST/INDICATION packet, or transmission of a CLEAR REQUEST packet, the “originator” value shall be the clearing cause and the “reason” value shall be the diagnostics value.

However, the values of originator and reason are a local matter in case of multiple attempts to set up an NC, failing finally.

11.7.2.4 Addresses

The contents of the address fields are determined by local operation.

11.7.2.5 User data

The clear user data field of X.25/PLP CLEAR REQUEST and CLEAR INDICATION packets is used to transfer BIS user data between users of the BIS.

NOTE – In CLEAR REQUEST and CLEAR INDICATION packets the Clear User Data field only exists, if the Fast Select facility was used in the corresponding CALL REQUEST or INCOMING CALL packet, which is an option for the BIS.

11.8 Data transfer phase – (Unqualified) data transfer service

11.8.1 Primitive/parameter and packet/field relationships

Table 59 shows the relationships between the primitives parameters and the packets/fields for the (unqualified) data transfer service.

TABLE 59/T.105

BIS/X.25 mapping for the (unqualified) data transfer service

BIS	X.25/PLP
PRIMITIVES BIS-N-DATA request BIS-N-DATA indication	PACKETS DATA DATA
PARAMETERS BIS-user-data	FIELDS User Data field Q-bit (Note) M-bit
NOTE – The value of the Q-bit is 0 in all packets of a complete data packet sequence.	

11.8.2 Procedures

11.8.2.1 Primitive/packet mapping

When an NL entity receives a BIS-N-DATA request primitive from a BIS user, it transmits a sequence of one or more DATA packets, known as an M-bit sequence (MBS), across the DTE/DXE interface. The number of DATA packets needed in an MBS depends on the amount of BIS-user-data and on the maximum data packet size (i.e. the maximum User Data field length of DATA packets) permitted at the DTE/DXE interface. All DATA packets but the last one have their M-bit set to 1. The last data packet has the M-bit set to 0. All DATA packets shall have the Q-bit set to 0.

When an NL entity receives an MBS with the Q-bit set to 0 in each DATA packet, it signals a BIS-N-DATA indication primitive to the BIS user.

If the Q-bit setting is not the same in all DATA packets of an MBS, the Network Layer entity shall reset the logical channel indicating the cause “DTE originated” and the diagnostics “Inconsistent Q-bit settings” (value = 83). Additionally the procedures for resetting an NC as defined below shall be followed.

11.8.2.2 User data

The User Data field of X.25/PLP DATA packets (Q-bit = 0) is used to transfer (unqualified) BIS user data between users of the BIS. The number and sequence of the octets contained in the BIS-user-data parameter remains unchanged by the network.

11.9 Data transfer phase – Qualified data transfer service (Q-bit = 1)

11.9.1 Primitive/parameter and packet/field relationships

Table 60 shows the relationships between the primitives parameters and the packets/fields for the qualified data transfer service.

TABLE 60/T.105

BIS/X.25 mapping for the qualified data transfer service

BIS	X.25/PLP
PRIMITIVES BIS-N-Q-DATA request BIS-N-Q-DATA indication	PACKETS DATA DATA
PARAMETERS BIS-user-data	FIELDS User Data field Q-bit (Note) M-bit
NOTE – The value of the Q-bit is 1 in all packets of a complete data packet sequence.	

11.9.2 Procedures

11.9.2.1 Primitive/packet mapping

When an NL entity receives a BIS-N-Q-DATA request primitive from a BIS user, it transmits a sequence of one or more DATA packets, known as an M-bit sequence (MBS), across the DTE/DXE interface. The number of DATA packets needed in an MBS depends on the amount of BIS user data and on the maximum data packet size (i.e. the maximum User Data field length of DATA packets) permitted at the DTE/DXE interface. All DATA packets but the last one have their M-bit set to 1. The last data packet has the M-bit set to 0. All DATA packets shall have the Q-bit set to 1.

When an NL entity receives an MBS with the Q-bit set to 1 in each DATA packet, it signals a BIS-N-Q-DATA indication primitive to the BIS user.

If the Q-bit setting is not the same in all DATA packets of an MBS, the Network Layer entity shall reset the logical channel indicating the cause “DTE originated” and the diagnostics “Inconsistent Q-bit settings” (value = 83). Additionally the procedures for resetting an NC as defined below shall be followed.

11.9.2.2 User data

The User Data field of X.25/PLP DATA packets (Q-bit = 1) is used to transfer qualified BIS user data between users of the BIS. The number and sequence of the octets contained in the BIS-user-data parameter remains unchanged by the network.

11.10 Data transfer phase – Reset service

11.10.1 Primitive/parameter and packet/field relationships

Table 61 shows the relationships between the primitives parameters and the packets/fields for the reset service.

TABLE 61/T.105

BIS/X.25 mapping for the reset service

BIS	X.25/PLP
PRIMITIVES	PACKETS
BIS-N-RESET request BIS-N-RESET indication	RESET REQUEST RESET INDICATION RESET REQUEST
BIS-N-RESET response BIS-N-RESET confirm	none (Note) none (Note)
PARAMETERS	FIELDS
Originator and Reason	Cause and Diagnostic Code fields
NOTE – There may be, but there need not be a relation to RESET packets. See also 11.10.2.2 below.	

11.10.2 Procedures

11.10.2.1 General Function

The reset service may be used:

a) by the BIS user to resynchronize the use of the NC; or

b) by the BIS provider to report detected loss of BIS user data unrecoverable within the BIS. All loss of user data which does not involve loss of the NC is reported this way.

All user data within the network and not delivered to the BIS user before completion of the reset service will be discarded by the BIS provider.

Valid sequences of primitives providing the reset service can be found in parts j) to m) of Figure 15.

11.10.2.2 Primitive/packet mapping

When an NL entity receives a BIS-N-RESET request primitive from a BIS user, it transmits a RESET REQUEST packet across the DTE/DXE interface. When the NL entity is ready to accept subsequent unqualified and qualified data from the BIS user, it signals an BIS-N-RESET confirm primitive. The issuing of this primitive may or may not be related to the completion of the X.25/PLP Reset procedure. Any unqualified data or qualified data received from the BIS user following the BIS-N-RESET confirm primitive is transmitted after completion of the X.25/PLP Reset procedure.

If an NL entity detects an error in the operation of the X.25/PLP for which its action is to reset the virtual circuit (e.g. after a sequence error), then it transmits a RESET REQUEST packet across the DTE/DXE interface. When the NL entity is ready to accept subsequent unqualified data and qualified data from the BIS user, it signals a BIS-N-RESET indication primitive. The issuing of this primitive may or may not be related to the completion of the X.25/PLP Reset procedure. Any unqualified data or qualified data received from the BIS user following the BIS-N-RESET response primitive is transmitted after completion of the X.25/PLP Reset procedure.

When an NL entity receives a RESET INDICATION or REQUEST packet, it signals a BIS-N-RESET indication primitive to the BIS user.

When a BIS-N-RESET response primitive is received from the BIS user, the NL entity shall be willing to accept subsequent unqualified data or qualified data from the BIS user used for transmission upon completion of the X.25/PLP Reset procedure.

During the Reset process, the following actions are taken by the NL entity with respect to the operation of the X.25/PLP:

a) For DATA or qualified DATA packets:

– those awaiting transmission may either be transmitted prior to transmitting a RESET packet or flushed from the queue of DATA packets awaiting transmission;

– those remaining in the transmit window when the Reset procedure is completed are flushed; and

– those that have been received prior to receiving a RESET packet but which do not constitute an entire MBS are flushed from the "MBS reassembly area".

b) The lower window edge for each direction of data transmission is set to 0 and subsequently transmitted DATA or qualified DATA packets are numbered starting from 0.

c) Any busy condition that had existed prior to the reset is considered not to exist any longer.

d) All timer and retransmission parameters relating to data transfer are set back to their initial value.

No action is required with respect to the provision of the BIS by an NL entity when it receives a RESET CONFIRMATION packet, or a RESET INDICATION/REQUEST packet in response to a RESET REQUEST packet (i.e. a reset collision). However, it shall then be capable of receiving subsequent DATA packets.

11.10.2.3 Originator and Reason

The combination of Originator and Reason parameters of the BIS-N-RESET primitives is mapped to/from the combination of Resetting Cause and Diagnostic Code fields.

In case of a BIS-N-RESET request (originator parameter absent), the NL entity inserts the value of "DTE originated" into the Resetting Cause field of the RESET REQUEST packet.

11.11 Data Transfer-Phase – Interrupt transfer service

11.11.1 Primitive/parameter and packet/field relationships

Table 62 shows the relationships between the primitives parameters and the packets/fields for the interrupt transfer service.

TABLE 62/T.105

BIS/X.25 mapping for the interrupt transfer service

BIS	X.25/PLP
PRIMITIVES BIS-N-INTERRUPT request BIS-N-INTERRUPT indication	PACKETS INTERRUPT INTERRUPT
PARAMETERS BIS-Interrupt-User-Data	FIELDS Interrupt User Data field

11.11.2 Procedures

11.11.2.1 Primitive/packet mapping

When an NL entity receives a BIS-N-INTERRUPT request primitive from a BIS user, it transmits one INTERRUPT packet across the DTE/DXE interface.

When an NL entity receives an INTERRUPT packet, it signals a BIS-N-INTERRUPT indication primitive to the BIS user.

11.11.2.2 User data

The Interrupt User Data field of X.25/PLP INTERRUPT packets is used to transfer BIS interrupt user data between users of the BIS. The number and sequence of the octets contained in the BIS-interrupt-user-data parameter remains unchanged by the network.

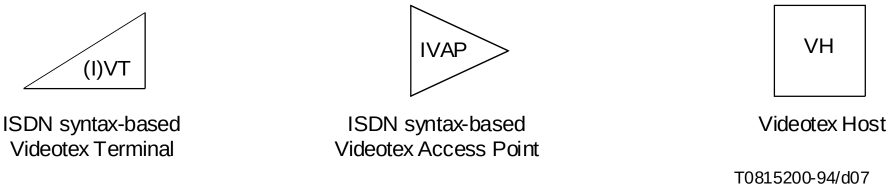
Annex A

Examples of configurations

(This annex does not form an integral part of this Recommendation)

A.1 Symbols

The following symbols in Figure A.1 are used in this annex:



(I)VT
IVAP
VH
ISDN syntax-based Videotex Terminal
ISDN syntax-based Videotex Access Point
Videotex Host
T0815200-94/d07

FIGURE A.1/T.105

Symbols

A.2 Access to PSTN/PSPDN systems using a terminal adaptor

These configurations are intended to access Videotex Services (VS), designed for use on the PSTN, via an ISDN. The services offered to the user should be identical to the service offered via a PSTN/PSPDN. As a consequence, the protocols to be used at the terminal side are those used for the access of PSTN/PSPDN Videotex Services.

These configurations are outside the scope of this Recommendation.

A.2.1 Analogue terminal adaptors

An ISDN may be utilized as an analogue telephone network (via the 3,1 kHz audio bearer service) using analogue terminal adaptors (TA). Both sides require modems. As shown in Figure A.2, the interworking unit between ISDN and PSTN can be used.

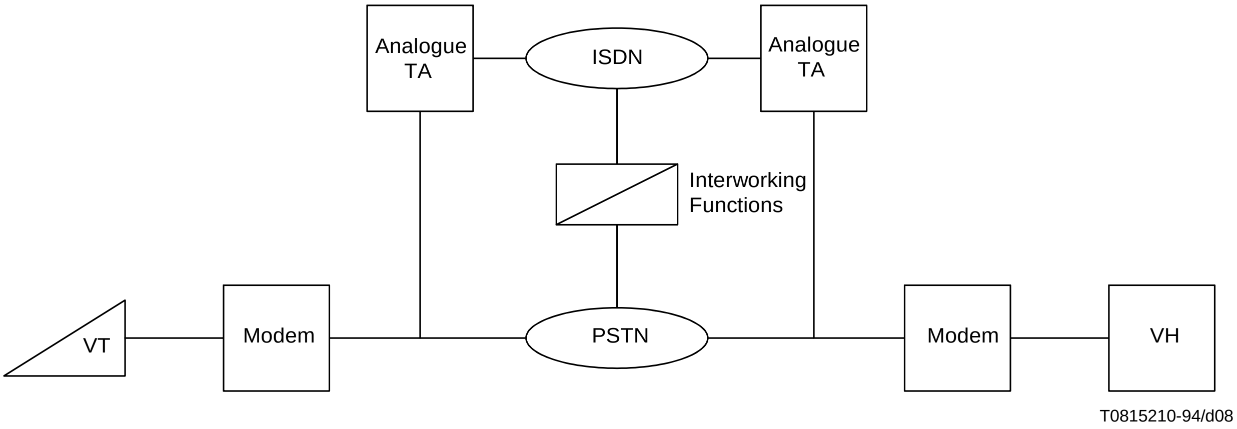


FIGURE A.2/T.105

Access of PSTN/PSPDN systems via analogue terminal Adaptors

A.2.2 Rate adapting terminal adaptors

Rate adapting terminal adaptors provide an end-to-end digital connection which transforms the digital 64 kbit/s link into an asynchronous V.24 link. No modems are needed, as shown in Figure A.3.

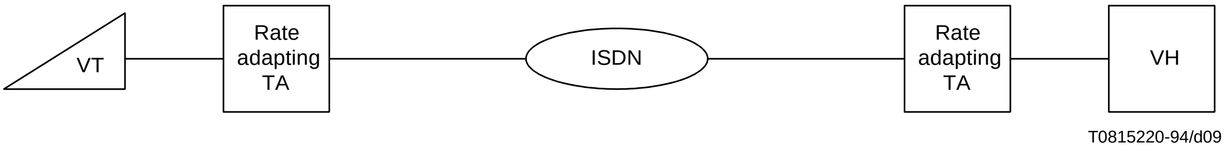


FIGURE A.3/T.105

Access of PSTN/PSPDN systems via rate adapting terminal adaptors

A.3 Connection to a VS via ISDN

The ISDN Videotex Terminal is connected directly to a Videotex Service via the ISDN. At the protocol level, one connection is established between the terminal and the Videotex Service (see Figures A.4 and A.5).

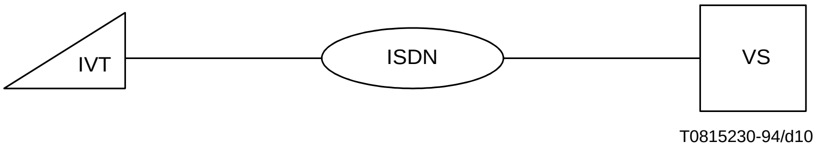


FIGURE A.4/T.105

Connection to a VS via ISDN

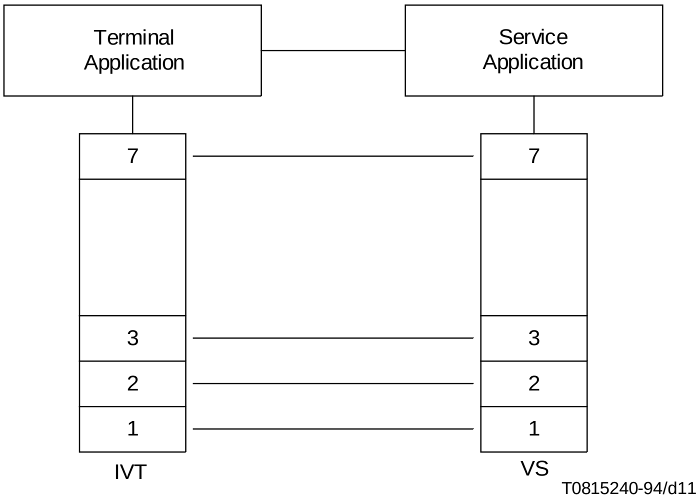


FIGURE A.5/T.105

Protocol stack for IVT-VS connection

A.4 Access to a VS via an ISDN/PSPDN

A Videotex Service is made available over the ISDN, using the interworking mechanisms between the ISDN and the PSPDN as defined in CCITT Recommendation X.31 for either case A or case B. These topologies imply that the packet interworking unit [an Access Unit (AU) in case A, a Packet Handler (PH) in case B] is addressed over the ISDN and the B-channel (or optionally the D-channel for case B) is used to establish a connection with the Videotex Service (see Figures A.6 and A.7).

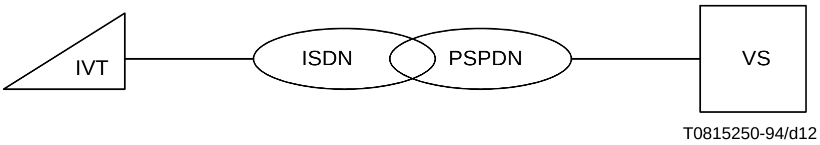


FIGURE A.6/T.105

Access to a VS via ISDN/PSPDN

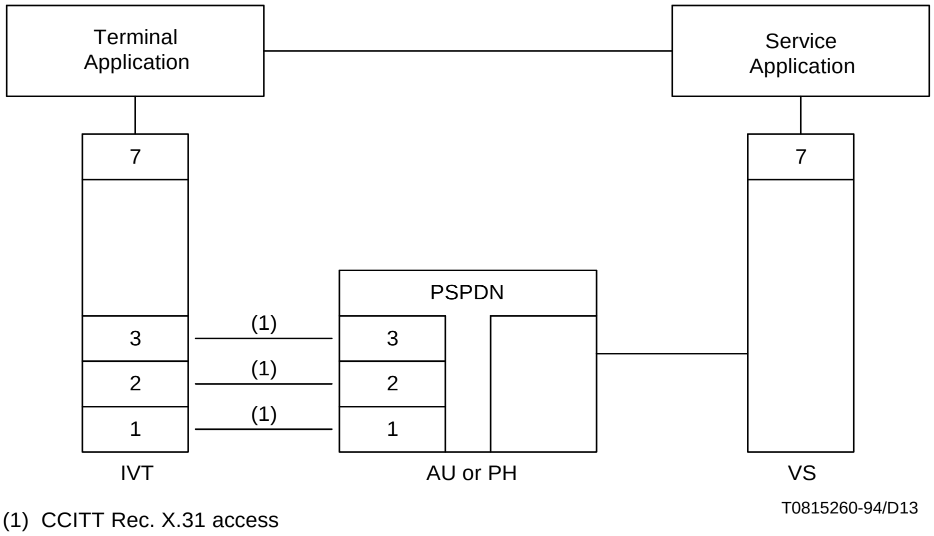


FIGURE A.7/T.105

Protocol stack for IVT-AU/PH-VS connection

A.5 Access to a VS via an IVAP

The terminal accesses a Videotex Service via an ISDN Videotex Access Point. The Host Access Network should be a PDN (generally a PSPDN) or an ISDN (see Figure A.8).

The connection establishment with the host does not have any influence on the protocol stack between the terminal and the IVAP. As far as the terminal is concerned, only a connection to the IVAP exists.

In terms of the protocol stack in the B-channel, there is an end-to-end connection on layer 3 between the two sides, with an Application Layer on top (see Figure A.9).

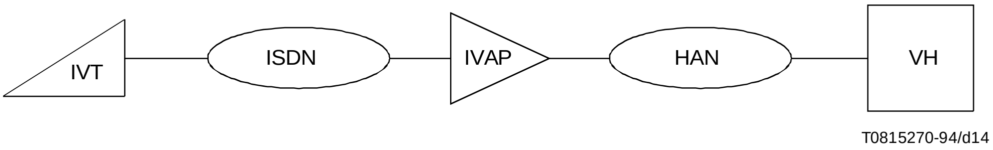


FIGURE A.8/T.105

Connection via an IVAP

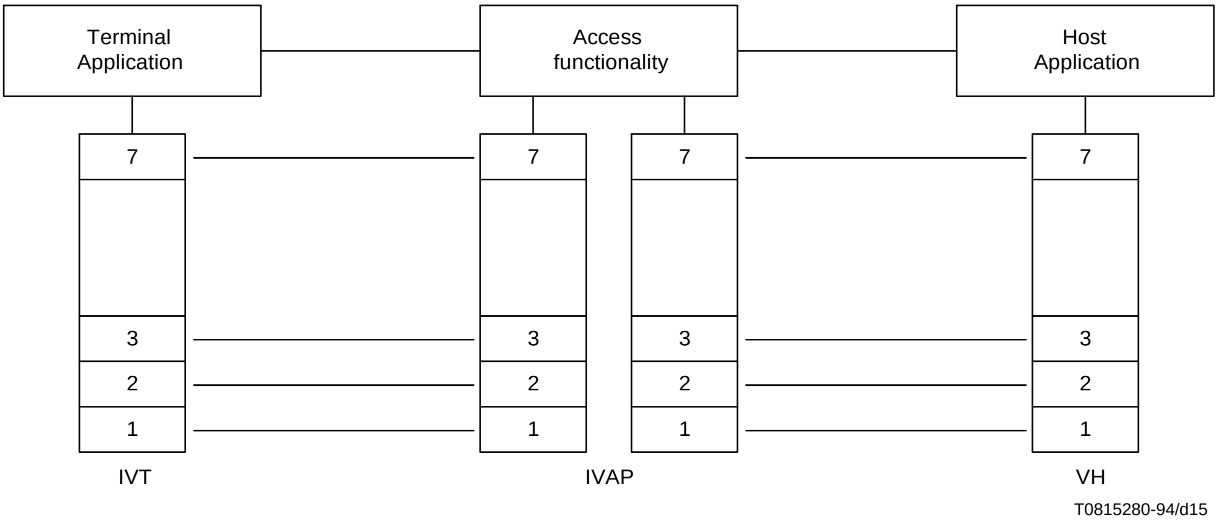


FIGURE A.9/T.105

Protocol stack for IVT-IVAP-VH connection

A.5.1 Service selection after a dialogue with the IVAP

The terminal is connected to an IVAP, which sets up a second connection to a server after some dialogue between the terminal and the IVAP. This is basically a refinement of the access to a VH (see Figure A.10).

A.5.2 Service selection using a VS identification

A B-channel is established between the terminal and the IVAP. On this B-channel, layer 2 and layer 3 are set up. The layer 3 CALL REQUEST carries the address or the name of the requested application. This information is used by the IVAP to establish an end-to-end dialogue between the terminal and the VH (see Figure A.11).

A.5.2.1 VS identified by a network address

The application is selected by its VH-address on the Host Access Network. This address is inserted in the layer 3 CALL REQUEST and interpreted by the IVAP to establish a connection with the VH.

EXAMPLE: Assume that the VH-address is “12345678”. This number is inserted in the layer 3 CALL REQUEST sent by the terminal to the IVAP.

NOTE – Some user data (e.g. user-identification) may be associated with the VH-address. It is conveyed transparently from the terminal to the VH.

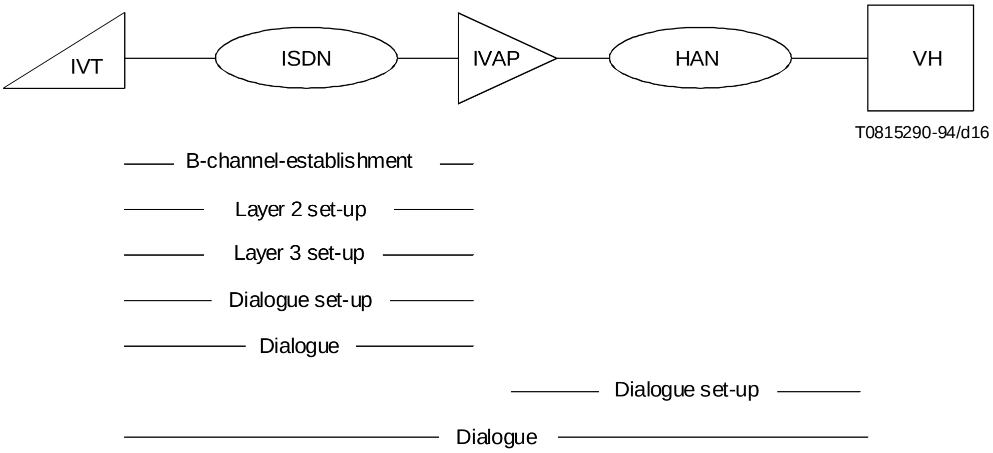


FIGURE A.10/T.105

Connection establishment after dialogue

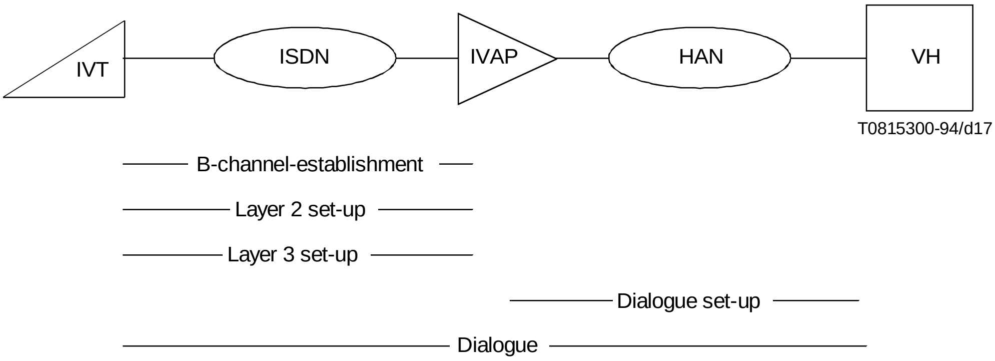


FIGURE A.11/T.105

Connection establishment without dialogue

A.5.2.2 VS identified by a name

The requested application is identified by a Videotex Service name. This name is inserted by the terminal in the layer 3 CALL REQUEST and interpreted by the IVAP to establish the appropriate connection.

EXAMPLE: Assume that the Videotex Service name is “ETSI”. This name is inserted in the layer 3 CALL REQUEST and translated by the IVAP into the VH-address “12345678”.

A.5.3 Service selection using ISDN supplementary services

In these configurations, ISDN supplementary services are used to convey service selection information before establishing a B-channel. This information can be used by the IVAP to select the appropriate application.

The usage of these mechanisms is independent of selection mechanisms which use B-channel protocols. Both mechanisms may be used together.

For more details on supplementary services, refer to Annex B (informative).

A.5.4 Distributed Videotex Application

See Figure A.12.

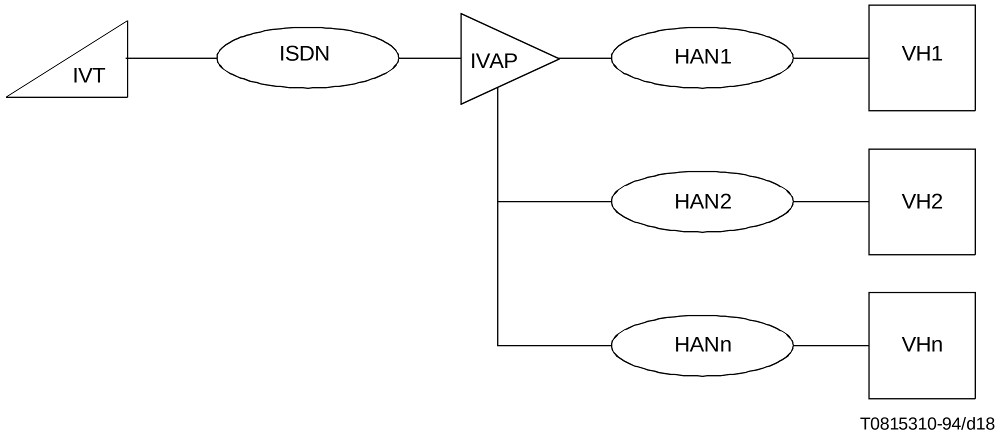


FIGURE A.12/T.105

Distributed Videotex Host configuration

NOTE – If one of the hosts is accessible only directly via a PSTN or PSPDN, the resulting configuration is outside the scope of this Recommendation.

A.5.4.1 Usage of a single VC between IVT and IVAP

The terminal is connected to the IVAP or VH1 using a VC on the B-channel. When the application requires the terminal to be connected to VH2, the IVAP should establish a VC to VH2 upon the reception of a specific request from VH1 (see Figure A.13).

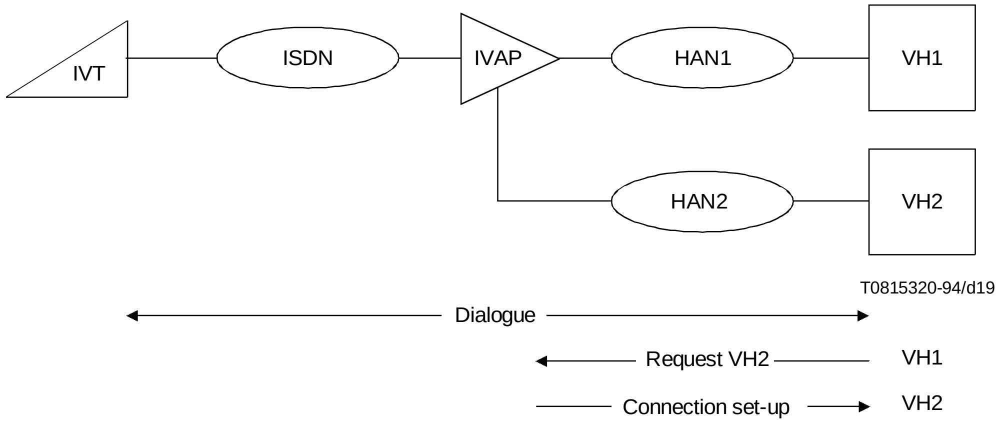


FIGURE A.13/T.105

Distributed hosts via a single VC

The terminal is linked to the IVAP via one single VC on one single B-channel, while the IVAP is connected to more than one VH. The IVAP should perform the multiplexing of informations received from VH1 and VH2 onto the one single VC to the terminal.

From a protocol standpoint at the terminal side, this configuration is exactly identical with the one described in A.4.

A.5.4.2 Establishment of the second VC by the IVT

The terminal is connected to the IVAP or VH1 using a VC on a B-channel. After having received a specific command from the IVAP or VH1, the terminal should establish a second VC on the same B-channel to VH2 (see Figure A.14).

The terminal should present all the informations which are received on the two different VCs. It should also be able to interpret a specific command requesting the setting up of a second VC.

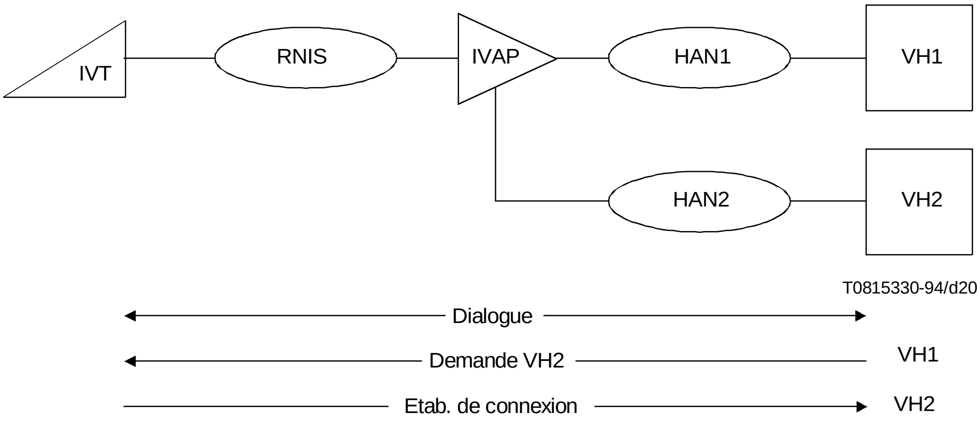


FIGURE A.14/T.105

Serveurs répartis via deux communications virtuelles établies par le terminal vidéotex

A.5.4.3 Establishment of the second VC by the IVAP

The terminal is connected to the IVAP or VH1 using a VC on the B-channel. When the application requires the terminal to be connected to VH2, the IVAP establishes a second VC connection to both VH2 and the terminal on the same B-channel. If the application is run on VH1, the second VC is established after having received a specific request from VH1 (see Figure A.15).

In this configuration, the terminal should present the informations which are received on two different VCs. It must also be able to accept a VC set-up.

A.6 Host to terminal call establishment

In all configurations described above, the initiative to establish the connection may also be taken by an IVAP. Therefore, a terminal should be able to respond to incoming calls. The topology of the network behind the IVAP is irrelevant (see Figure A.16).

After establishment of layer 3, the same procedures apply as for all above configurations, except for the connection establishment in conjunction with a direct service selection and the service selection using ISDN supplementary services.

A.7 Terminal to terminal communication

If the IVAP in the above Figure A.16 is replaced by another terminal, the resulting configuration allows for a direct terminal to terminal communication, as shown in Figure A.17.

After establishment of layer 3, none of the service selection procedures are applicable. The Application Layer protocol does not contain any special support for this configuration.

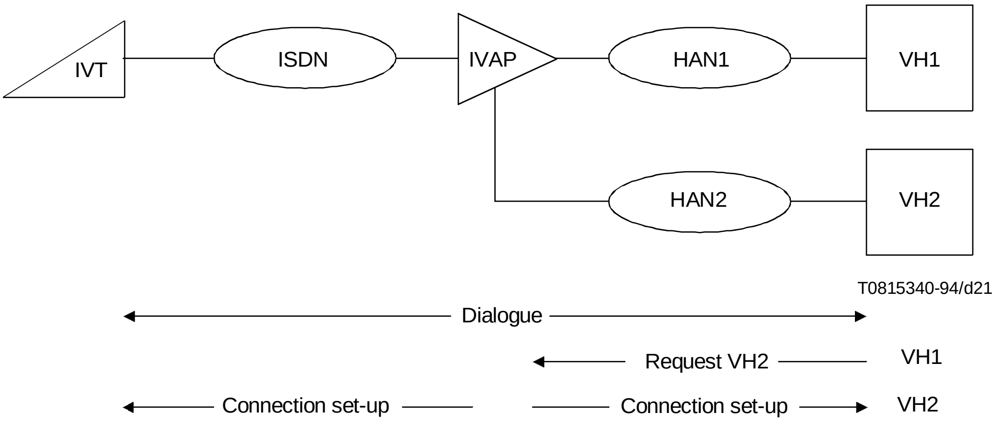


FIGURE A.15/T.105

Distributed hosts via two VCs initiated by the IVAP

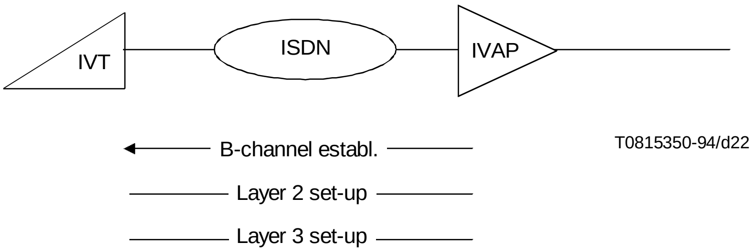


FIGURE A.16/T.105

Host to terminal connection establishment

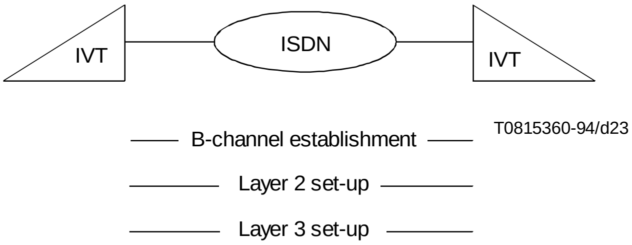


FIGURE A.17/T.105

Terminal to terminal communication

Annex B

Usage of supplementary services

(This annex does not form an integral part of this Recommendation)

This annex gives an overview of the supplementary services which may be used in Syntax-based Videotex in case of an ISDN. The utilization of any supplementary service is optional.

NOTE – The use of supplementary services applies only to communications in the circuit-switched mode using a DTE/DTE connection in the Network Layer. For all other cases, the use of supplementary services is under study.

B.1 User-to-User Signalling (UUS)

The UUS supplementary service allows an ISDN user to send/receive a limited amount of information to/from another ISDN user over the D-channel. This information is passed transparently through the network. The message size is restricted to 128 octets.

The user can transfer user-to-user information (UUI) in different phases of the call depending on the service(s) to which the user subscribes. For the ISDN Syntax-based Videotex, only Service 1 shall be applicable. Service 1 provides for the transfer of UUI during the setup and clearing phases of a call with UUI embedded within call control messages.

A calling party may be subscribed to UUS. The terminal should be able to enter a UUI in the SETUP message. The specification how to handle the UUS information is outside the scope of this Recommendation.

B.2 Supplementary services for addressing

B.2.1 Multiple Subscriber Number (MSN)

MSN allows for more than one subscriber number being allocated to a single basic access interface. By inspecting the called number on the D-channel, terminals can decide whether they should accept a call. The interpretation of the number is up to the terminals.

A calling party may be subscribed to MSN. This does not have any impact on the terminal-network interface unless the user is subscribed to CLIP. In this case, a terminal has to insert its subscriber number (one of the numbers assigned to the network interface) in the messages conveyed to the network.

B.2.2 Direct-Dialling-In (DDI)

DDI is intended to subaddress a specific terminal (or interface) on an integrated services private branch exchange (ISPBX), which is connected to the ISDN via a primary rate interface.

A calling party may be subscribed to DDI. This does not have any impact on the terminal-network interface unless the user is subscribed to CLIP. In this case, a terminal has to insert its subscriber number (its local extension) in the messages conveyed to the network.

B.2.3 Subaddressing (SUB)

Subaddressing (SUB) is a mechanism to convey a number of digits to the called party which may use this information for terminal selection. The maximum size of the subaddress is 20 octets. However, for a certain period of time, the size of the subaddress can be limited to a maximum of 4 octets either within certain networks or between networks.

A calling party may be subscribed to SUB. The terminal should be able to enter a subaddress in the SETUP message.

B.2.4 Call Deflection (CD)

A called party may be subscribed to CD. It can use the information conveyed through the UUS 1 supplementary service messages to deflect a call to another address. The specification how to handle the UUS information is outside the scope of this Recommendation.

B.3 Supplementary services for identification

B.3.1 Calling Line Identification Presentation (CLIP)

A calling party may be subscribed to CLIP. If the calling party is subscribed to CLIP, the calling line number is sent to the called party as one component of the SETUP message. Generally, the calling line identification information should be inserted by the network. In specific cases, however, terminals can insert the calling line identification information in the SETUP message. This feature is mainly used on network interfaces subscribed to MSN or DDI.

B.3.2 Calling Line Identification Restriction (CLIR)

A calling party may be subscribed to CLIR. In case the user is subscribed to permanent mode, no calling line identification is transmitted to the called party. If the user is subscribed to temporary mode, he can indicate to the network whether the calling line identification should be transmitted to the called party or not by inserting a presentation indicator in the SETUP message.

Annex C

Terminal Function basic state

(This annex forms an integral part of this Recommendation)

The initial state of a Terminal Function after the successful establishment of a network connection, prior to the exchange of any further protocol elements, is called basic state. In this basic state, parameters and switches shall have the default values as defined below.

C.1 X.3 parameter default values

The default values for all X.3 parameters which are used by the Syntax-based Videotex are given in Table C.1.

TABLE C.1/T.105

CCITT Recommendation X.3 [3] Syntax-based videotex profile

Parameter reference number	Parameter description	Default value
2	Echo (M)	0
3	Selection of data forwarding character(s) (M)	0
4	Selection of idle timer delay (M)	1
23	Size of input field (O)	0
24	End of frame signals (O)	32
25	Extended data forwarding signals (O)	0
26	Display interrupt (O)	0
28	Diacritic character editing (O)	0
29	Extended echo mark (O)	0

M Mandatory
O Optional

NOTE – No default value for parameter reference 11 is defined.

C.2 SBV_DFK default list

The SBV_DFK default list shall be empty.

C.3 SBV_TPD default state

The SBV_TPD regime shall not be established.

C.4 SBV_Escape default state

When implemented, any process related to an SBV_Escape subservice shall be inactive or shall be in its idle state.

Annex D

The SBV_Escape service

(This annex does not form an integral part of this Recommendation)

D.1 Service description

D.1.1 Bulk Update

In the sense of CCITT Recommendation F.300 [1] “a Videotex service facility is an Application Layer implementation in a Videotex service, providing a specific, clearly defined facility to Videotex users” (see 2.2.1).

One of the several possible service facilities is named “Videotex transaction (...) which allows users to create and/or modify information stored in a data base” (see 2.2.6). Another facility is named “Videotex data processing (...) which allows the user to employ processing and storage capacity at the host computer” (see 2.2.6).

The information elements to be handled by these service facilities may be a Videotex frame (see 2.4.13) for the first case or a Videotex System field (see 2.4.16) for the latter case.

The SBV_Escape subservices Bulk Update as defined in this clause provide for protocol elements to use these service facilities/applications for both ISDN and PSTN access in a given Videotex Service.

The Bulk Update subservices are based on a master/slave handshake procedure defined for a PSTN access which allows for the transmission of 8-bit coded data blocks under the control of the Videotex Application. As a consequence, this subservice has to follow some rules which are not conclusive for a sole ISDN access.

D.2 Procedures

D.2.1 Bulk Update

Both Terminal Function and Access Function may use the Bulk Update subservices according to the following rules.

NOTE – Additional rules concerning possible sequences of Bulk Update protocol elements are under the responsibility of the service facility provider (see CCITT Recommendation F.300 [1], 2.1.3).

A data block can be transmitted in portions using the BULK_DATA subservice. The portion size depends on the Videotex Service facility attached.

The last data portion is conveyed using either one of the subservices BULK_LAST or BULK_EOT. This data portion may be the only one or it may even be empty.

BULK_LAST indicates the last portion of the data block to be transmitted.

BULK_EOT indicates in addition that no further data blocks will follow.

BULK_CAN may be used to abort the transaction at any given time.

D.3 Coding

The SBV_Escape subservices are identified by values of the Service-ID parameter as defined in Table D.1.

TABLE D.1/T.105

Values for the Service-ID parameter

SBV_Escape Subservice	Service-ID
BULK_DATA	8/0
BULK_LAST	8/1
BULK_EOT	8/2
BULK_CAN	8/3
NOTE – Telecommunication Standardization Sector Study Group 8 should be in charge of allocating code values of the SBV_Escape subservices.	

Annex E

Extended data forwarding signals

(This annex does not form an integral part of this Recommendation)

The CCITT Recommendation X.3 [3] extended parameter reference 25, which is introduced in this Recommendation, defines a number of additional forwarding conditions.

These extended data forwarding signals are not based upon a Recommendation, but originate from national and de facto standards. The following overview gives some background to the introduction of these conditions.

NOTE – Information providers should only use forwarding conditions which are related to the system/profile being used.

Sequence of two characters starting with character 1/3

MINITEL terminals used in the TELETEL and other profile 2 compatible networks issue these sequences when function keys are pressed.

Character 1/10

This character (Data Collection Terminator, DCT) is used in several profile 1 orientated services to indicate to the access function, that the filling of the current form is terminated.

Character 1/12

This character (Terminator, TER) is used in several profile 1 orientated services to indicate to the access function, that the filling of the current field is terminated.

Character 2/3

This character (“#”) is used in several PRESTEL (profile 3) orientated services to indicate to the access function, that the filling of the current field is terminated.

Character 5/15

This character (“_”, but on various PRESTEL terminals displayed as “#”) is used in several PRESTEL (profile 3) orientated services to indicate to the access function, that the filling of the current field is terminated.

Sequence of the two characters ESC “J” (1/11 4/10)

This sequence (the END BOX sequence in a 7-bit coding structure) is used in several PRESTEL (profile 3) orientated services to indicate to the access function, that the filling of the current form is terminated.

Sequence of three characters starting with the two-character sequence 2/10 3/0

These sequences are used in several PRESTEL (profile 3) orientated services to invoke special service functions (redisplay a frame, retrieve updated information, advice of charge). They do not have a specific termination character, but a fixed command length.

Annex F

BIS constraints on ISO 8208 PICS

(This annex forms an integral part of this Recommendation)

F.1 Introduction

This annex contains a list of constraints on the status information of the ISO/IEC 8208:1990/Am.3 [16] PICS proforma, posed by the requirements of the Bearer Independent Service as defined in clause 11. The proforma is intended to be a supplement to the ISO/IEC 8208:1990/Am.3 [16] PICS proforma and all the general information (e.g. fill-in information for the client and phrasing of conditions) is valid for the constraints Tables as well.

F.2 Tables structure

Each of the following Tables F.1 to F.5 contains the following columns:

1) “Item”

   Used to identify the conformance statement. Only those items from the ISO/IEC 8208:1990/Am.3 [16] PICS appear, for which a status change is required.

2) “Feature”

   Used to give a short description of the item.

3) “8208 status”

   Original status of the item.

4) “8208 Reference”

   Related subclause(s) of ISO/IEC 8208 [15].

5) “BIS Status”

   Used to indicate the new status of the statement. The status values are phrased in the same way as in the original PICS.

6) “Support

   (Yes/No)”

   The client should fill in the proforma in this column, using a Yes or No for each item.

F.3 Constraints Tables

F.3.1 Virtual circuit types

See Table F.1.

TABLE F.1/T.105

Item	Feature	8208 Status	8208 Reference	References	BIS Status	Support Yes/No
Vp	PVC	O.1		11.2	X	
Vs	VC	O.1	5	11.2	M	

F.3.2 Call setup

See Table F.2.

TABLE F.2/T.105

Item	Feature	8208 Status	8208 Reference	References	BIS Status	Support Yes/No
S1	Are outgoing VCs supported?		5.2.1 5.2.5			
S1a	– Fast Select, no restriction	O	5.2.4	11.4.3	O.1	
S1b	– Fast Select, restriction	O	13.16	11.4.3	X	
S1c	– no Fast Select	O	5.2.4	11.4.3	O.1	

F.3.3 Call clearing

See Table F.3.

TABLE F.3/T.105

Item	Feature	8208 Status	8208 Reference	References	BIS Status	Support Yes/No
	Is call clearing supported as:		5.5.4			
C1	– response to indication of clearing	O	5.5.2	11.7	M	
C2a	– aborting an outgoing VC attempt	O	5.4	11.7	M	
C2c	– originating clearing of an established VC	O	5.5.1	11.7	M	

F.3.4 Resetting of logical channels

See Table F.4.

TABLE F.4/T.105

Item	Feature	8208 Status	8208 Reference	References	BIS Status	Support Yes/No
	Is resetting supported:		8			
RSi	– as initiator	O	8.1	11.10	M	
RSr	– as responder	O	8.2	11.10	M	

F.3.5 Normal data transfer and flow control

See Table F.5.

TABLE F.5/T.105

Item	Feature	8208 Status	8208 Reference	References	BIS Status	Support Yes/No
	Are the following supported:					
DS1	– Sending DATA packets	O	6.6	11.8; 11.9	M	
DS5a	– Sending Q = 0 in DATA packets	O.10	6.6	11.8	M	
DS5b	– Sending Q = 1 in DATA packets	O.10	6.6	11.9	M	
DR1	– Receiving DATA packets	O	6.4; 6.5	11.8; 11.9	M	
DR4b	– Receiving M = 1 in DATA packets	O	6.4	11.8; 11.9	M	
DR5a	– Receiving Q = 0 in DATA packets	O.11	6.6	11.8	M	
DR5b	– Receiving Q = 1 in DATA packets	O.11	6.6	11.9	M	