INTERNATIONAL TELECOMMUNICATION UNION

# ITU-T X.212

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

## OPEN SYSTEMS INTERCONNECTION

## SERVICE DEFINITIONS

---

# DATA LINK SERVICE DEFINITION FOR OPEN SYSTEMS INTERCONNECTION FOR CCITT APPLICATIONS

## ITU-T Recommendation X.212

(Extract from the *Blue Book*)

---

**NOTES**

1 ITU-T Recommendation X.212 was published in Fascicle VIII.4 of the *Blue Book*. This file is an extract from the *Blue Book.* While the presentation and layout of the text might be slightly different from the *Blue Book* version, the contents of the file are identical to the *Blue Book* version and copyright conditions remain unchanged (see below).

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

© ITU 1988, 1993

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

**Recommendation X.212**

**DATA LINK SERVICE DEFINITION FOR OPEN SYSTEMS INTERCONNECTION FOR CCITT APPLICATIONS[1)]**

*(Melbourne, 1988)*

The CCITT,

*considering*

(a) that Recommendation X.200 defines the Reference Model of Open Systems Interconnection for CCITT applications;

(b) that Recommendation X.210 specifies the OSI Layer Service Definition Conventions for describing the services of layers of the OSI Reference Model,

*unanimously recommends*

(1) that the scope and field of application of the Open Systems Interconnection Data Link Service Definition are given in § 1;

(2) that the general requirements of the Open Systems Interconnection Data Link Service are given in Part 1;

(3) that the connection-mode Data Link Service is defined in Part 2;

(4) that the connectionless-mode Data Link Service is defined in Part 3.

CONTENTS


0 *Introduction*

1 *Scope and field of application*

2 *References*

*Part One* - General

3 *Definitions*

3.1 OSI Reference Model Definitions

3.2 Service Conventions Definitions

3.3 Data Link Service Definitions

4 *Abbreviations*

5 *Conventions*

5.1 General Conventions

5.2 Parameters

6 *Overview of the Data Link Service*

7 *Classes, types and provided options of Data Link Service*

*Part Two* - Definition of connection-mode service

8 *Features of the connection-mode Data Link Service*

9 *Model of the connection-mode Data Link Service*


---

1) Recommendation X.212 and ISO 8886 [Information Processing Systems - Data Communications - Data Link Service Definition] were developed in close collaboration and are technically aligned, except for the differences noted in Appendix I.

9.1 DLC Endpoint Connection Identification

9.2 Model of a Data link connection

10 *Quality of connection-mode Data Link Service*

10.1 Determination of QOS for Connection-mode service

10.2 Definition of QOS Parameters

11 *Sequence of primitives*

11.1 Concepts used to Model the Connection-mode Data Link Service

11.2 Constraints on Sequence of Primitives

12 *Connection establishment phase*

12.1 Function

12.2 Types of Primitives and Parameters

12.3 Sequence of Primitives

13 *Connection release phase*

13.1 Function

13.2 Types of Primitive and Parameters

13.3 Sequence of Primitives when Releasing an established DLC

13.4 Sequence of Primitives in a DLS User rejection of DLC establishment attempt

13.5 Sequence of Primitives in a DLS Provider rejection of a DLC establishment attempt

13.6 Sequence of Primitives in a DLS User abort of a DLC connection attempt

14 *Data transfer phase*

14.1 Data transfer

14.2 Reset Service

*Part Three* - Definition of connectionless-mode primitives

15 *Features of the connectionless-mode Data Link Service*

16 *Model of the connectionless-mode Data Link Service*

16.1 Model of a Data-link-connectionless-mode transmission

17 *Quality of connectionless-mode service*

17.1 Determination of QOS for connectionless-mode service

17.2 Definition of connectionless-mode QOS Parameters

18 *Permitted sequence of connectionless-mode primitives*

19 *Data transfer*

19.1 Functions

19.2 Types of primitives and parameters

19.3 Sequence of primitives

*Appendix I* Differences between CCITT and ISO texts

*Appendix II* The relationship between the two types of Data Link Service

*Appendix III* Use of the X.25 LAPB compatible DTE data link procedures to provide the connection-mode data link service for open systems interconnection for CCITT applications.

## 0 Introduction

0.1 *About this Recommendation*

This Recommendation is one of a set of Recommendations produced to facilitate the interconnection of information processing systems. It is related to other Recommendations in the set as defined by Recommendation X.200, Reference Model of Open Systems Interconnection for CCITT Applications. The reference model described by Recommendation X.200 subdivides the area of standardization for Open Systems Interconnection (OSI) into a series of layers of specification, each of a manageable size.

This Recommendation defines the services provided by the Data Link Layer to the Network Layer at the boundary between the Data Link and Network Layers of the OSI Reference Model. It provides for the designers of network protocols a definition of the Data Link Service existing to support the network protocol and for the designers of Data Link Protocols a definition of the services to be made available through the action of the Data Link Protocol over the underlying service. The relationship is illustrated in Figure 1/X.212.

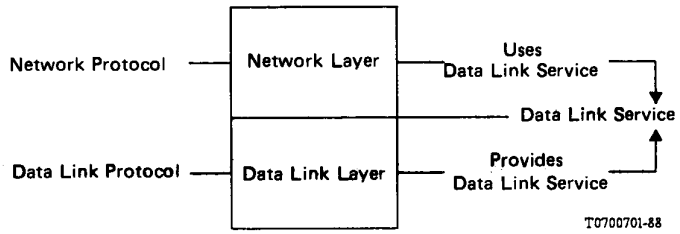


FIGURE 1/X.212

**Relationship of this Recommendation to other OSI Recommendations**

Throughout the set of OSI Recommendations, the term "service" refers to the abstract capability provided by one layer of the OSI Reference Model to the layer immediately above. Thus, the Data Link Service defined in this document is a conceptual architectural service, independent of administrative divisions.

## 1 Scope and field of application

This Recommendation defines the OSI Data Link Service in terms of:

a) the primitive actions and events of the service;

b) the parameters associated with each primitive action and event, and the form that they take; and

c) the interrelationship between, and the valid sequences of these actions and events.

The principal objective of this Recommendation is to specify the characteristics of a conceptual Data Link Service and thus, supplement the OSI Reference Model in guiding the development of Data Link Protocols.

This Recommendation does not specify individual implementation or products, nor does it constrain the implementation of Data Link entities and interfaces within an information processing system.

There is no conformance of equipment to this Data Link Service Definition Recommendation. Instead, conformance is achieved through implementation of conforming Data Link Protocols that fulfil the Data Link Service defined in this Recommendation.

## 2 References

Recommendation X.200 - Reference Model of Open Systems Interconnection for CCITT Applications (see also ISO 7498).

Recommendation X.210 - OSI Layer Service Definition Conventions (see also ISO TR8509).

## 3 Definitions

### 3.1 *OSI Reference Model Definitions*

This Recommendation is based on the concepts developed and makes use of the following terms defined in Recommendation X.200:

a) Data link entity

b) Data Link Layer

c) Data-link-Service

d) Data-link-service-access-point

e) Data-link-service-access-point-address

f) Data-link-service-data-unit

g) Reset.

### 3.2 *Service Conventions Definitions*

This Recommendation makes use of the following terms defined in Recommendation X.210, as they apply to the Data Link Layer:

a) Data-link Service User

b) Data Link Service Provider

c) Primitive

d) Request

e) Indication

f) Response

g) Confirm

3.3 *Data Link Service Definitions*

This Recommendation makes use of the following terms:

a) **data-link-connection**

An association established by a Data Link Layer between two or more Data Link Service users for the transfer of data, which provides explicit identification of a set of Data Link data transmissions and agreement concerning the Data Link data transmission services to be provided for the set.

*(Note* - This definition clarifies the definition given in X.200.)

b) **data-link-connection-mode data transmission**

The transmission of a Data-link-service-data-unit within the context of a Data-link-connection that has been previously established.

c) **data-link-connectionless-mode data transmission**

The transmission of a Data-link-service-data-unit not in the context of a Data-link-connection and not required to maintain any logical relationship among multiple invocations.

## 4 Abbreviations

DL Data Link

DLC Data-link-connection

DLL Data Link Layer

DLS Data Link Service

DLSAP Data-link-service-access-point

DLSDU Data-link-service-data-unit

OSI Open Systems Interconnection

QOS Quality of Service

## 5 Conventions

5.1 *General conventions*

This Recommendation uses the descriptive conventions given in Recommendation X.210.

The service model, service primitives, and time-sequence diagrams used are entirely abstract descriptions; they do not represent a specification for implementation.

5.2 *Parameters*

Service primitives, used to represent service user/service provider interactions (Recommendation X.210), convey parameters which indicate information available in the user/provider interaction.

The parameters which apply to each group of Data Link Service primitives are set out in tables in sections 12 to 14 and 19. Each "X" in the tables indicates that the primitive labelling the column in which it falls may carry the parameter labelling the row in which it falls.

Some entries are further qualified by items in brackets. These may be:

a) A parameter specific constraint:

(=) indicates that the value supplied in an indication or confirm primitive is always identical to that supplied in a previous request or response primitive issued at the peer service-access-point

b) Indication that some note applies to the entry:

(see note X) indicates that the referenced Note contains additional information pertaining to the parameter and its use.

In any particular interface, not all parameters need be explicitly stated. Some may be implicitly associated with the DLSAP at which the primitive is issued.

## 6 Overview of the Data Link Service

The DLS provides for the transparent and reliable transfer of data between DLS users. It makes invisible to these DLS users the way in which supporting communications resources are utilized to achieve this transfer.

In particular, the DLS provides for the following:

a) Independence of underlying Physical Layer - the DLS relieves DLS users from all concerns regarding which configuration is available (e.g. point-to-point connection) or which physical facilities are used (e.g. half-duplex transmission).

b) Transparency of transferred information - the DLS provides for the transparent transfer of DLS user-data. It does not restrict the content, format or coding of the information, nor does it ever need to interpret its structure or meaning.

c) Reliable transfer of data - the DLS relieves the DLS user from loss, insertion, corruption or, if requested, misordering of data which may occur. In some cases of unrecoverable errors in the Data Link Layer, duplication or loss of DLSDUs may occur.

*Note* - Detection of duplicate or lost DLSDUs may be performed by DLS users.

d) Quality of Service selection - the DLS makes available to DLS users a means to request and to agree upon a quality of service for the transfer of data. QOS is specified by means of QOS parameters representing characteristics such as throughput, transit delay, accuracy and reliability.

e) Addressing: The DLS allows the DLS user to identify itself and to specify the DLSAP to which a DLC is to be established whenever more than two DLSAPs are supported by the DLS provider. Data Link addresses have only local significance within a specific Data Link configuration over a single transmission medium (point-to-point or multi-point physical connection) or a group of parallel transmission media (multi-link or splitting function). Therefore it is not appropriate to define a global addressing structure.

*Note* - The DLS is required to differentiate between the individual systems that are physically or logically connected to a multipoint data link and to differentiate between connections when the data link layer includes a multiplexing function. For commonality with other Service definitions, this mechanism is referred to as addressing and the objects used to differentiate between systems are referred to as addresses.

## 7 Classes and types of Data Link Service

There are no distinct classes of Data Link Service defined. There are two types of Data Link Service:

a) a connection-mode service (defined in Part 2); and

b) a connectionless-mode service (defined in Part 3).

When making reference to this Recommendation, a user or provider of Data Link Service shall state which types of service it expects to use or provide.

## 8 Features of the connection-mode Data Link Service

The DLS provides the following features to the DLS user:

a) the means to establish a DLC with another DLS user for the purpose of exchanging DLSDUs;

b) the establishment of an agreement between the initiating DLS user and the DLS provider for a certain QOS associated with each DLC;

c) the means of transferring DLSDUs of restricted length on a DLC. The transfer of DLSDUs is transparent, in that the boundaries of DLSDUs and the contents of DLSDUs are preserved unchanged by the DLS, and there are no constraints on the DLSDU content imposed by the DLS;

*Note* - The length of a DLSDU may be limited because of internal mechanisms employed by the data-link-protocol (see Recommendation X.200 § 7.6.3.2).

d) the means by which the receiving DLS user may flow control the rate at which the sending DLS user may send DLSDUs;

e) the means by which a DLC can be returned to a defined state and the activities of the two DLS users synchronized by use of a Reset service element;

*f)* the unconditional, and therefore possibly destructive, release of a DLC by either of the DLS users or by the DLS provider.

## 9 Model of the connection-mode Data Link Service

This Recommendation uses the abstract model for a layer service defined in section 4 of Recommendation X.210. The model defines the interactions between the DLS users and the DLS provider which take place at the two DLSAPs. Information is passed between the DLS user and the DLS provider by service primitives, which may convey parameters.

### 9.1 *DLC Endpoint Connection Identification*

If a DLS user needs to distinguish among several DLCs at the same DLSAP, then a local connection endpoint identification mechanism must be provided. All primitives issued at such a DLSAP within the context of a DLC would be required to use this mechanism to identify this DLC. Such an implicit identification is not described in this Recommendation.

### 9.2 *Model of a Data-link-connection*

Between the two endpoints of a DLC, there exists a flow control function that relates the behaviour of the DLS user receiving data to the ability of the other DLS user to send data. As a means of specifying this flow control feature and its relationship with other capabilities provided by the connection-mode DLS the queue model of a DLC, which is described in the following sections, is used.

This queue model of a DLC is discussed only to aid in the understanding of the end-to-end service features perceived by DLS users. It is not intended to serve as a substitute for a precise, formal description of the DLS, nor as a complete specification of all allowable sequences of DLS primitives. (Allowable primitive sequences are specified in section 11. See also the note below.) In addition, this model does not attempt to describe all the functions or operations of DL entities that are used to provide the DLS. No attempt to specify or constrain DLS implementations is implied.

*Note* - The internal mechanisms which support the operation of the DLS are not visible to the DLS user. In addition to the interactions between service primitives described by this model (e.g. the issue of a DL-Reset request at an DLSAP may prevent the receipt of a DL-DATA indication, corresponding to a previously issued DL-DATA request, by the peer DLS user) there may also be:

a) constraints applied locally on the ability to invoke primitives;

b) service procedures defining particular sequencing constraints on some primitives.

9.2.1 *Queue model concepts*

The queue model represents the operation of a DLC in the abstract by a pair of queues linking the two DLSAPs. There is one queue for each direction of information flow (see Figure 2/X.212).

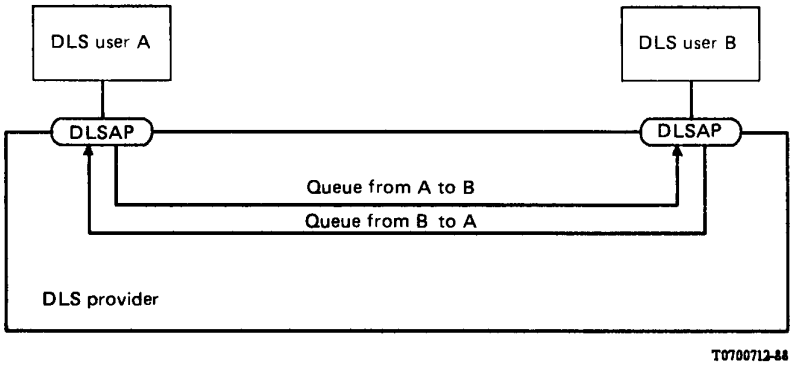


FIGURE 2/X.212

**Queue model of a DLC**

Each queue represents a flow control function in one direction of transfer. The ability of a DLS user to add objects to a queue will be determined by the behaviour of the other DLS user in removing objects from the queue. Objects are entered or removed from the queue as a result of interactions at the two DLSAPs.

The pair of queues is considered to be available for each potential DLC.

The following objects may be placed in a queue by a DLS user (see sections 12-14):

a) a connect object, representing a DL-CONNECT primitive and its parameters;

b) a data object, representing a DL-DATA primitive and its parameters;

c) a reset object, representing a DL-RESET primitive and its parameters; and

d) a disconnect object, representing a DL-DISCONNECT primitive and its parameters.

The following objects may be placed in a queue by the DLS-provider (see sections 12-14):

1) a reset object, representing a DL-RESET primitive and its parameters;

2) a synchronization mark object (see § 9.2.4); and

3) a disconnect object, representing a DL-DISCONNECT primitive and its parameters.

The queues are defined to have the following general properties:

i) a queue is empty before a connect object has been entered and can be returned to this state, with loss of its contents, by the DLS provider;

ii) objects are entered into a queue by the sending DLS user, subject to control by the DLS provider. Objects may also be entered by the DLS provider;

iii) objects are removed from the queue, under the control of the receiving DLS user;

iv) objects are normally removed in the same order that they were entered (however, see § 9.2.3); and

v) a queue has a limited capacity, but this capacity is not necessarily either fixed or determinable.

9.2.2 *DLC Establishment*

A pair of queues is associated with a DLC between two DLSAPs when the DLS provider receives a DL-CONNECT request primitive at one of the DLSAPs, and a connect object is entered into one of the queues. From the standpoint of the DLS users of the DLC, the queues remain associated with the DLC until a disconnect object representing a DL-DISCONNECT primitive is either entered or removed from the queue.

DLS user A, who initiates a DLC establishment by entering a connect object representing a DLCONNECT request primitive into the queue from DLS user A to DLS user B, is not allowed to enter any other object, other than a disconnect object, into the queue until after the connect object representing the DLCONNECT confirm primitive has been removed from the DLS user B to DLS user A queue. In the queue from DLS user B to DLS user A objects can be entered only after DLS user B has entered a connect object representing a DL-CONNECT response primitive.

The properties exhibited by the queues while the DLC exists represent the agreements reached among the DLS users and the DLS provider during this connection establishment procedure concerning QOS.

9.2.3 *Data transfer*

Flow control on the DLC is represented in this queue model by the management of the queue capacity, allowing objects to be added to the queues. The addition of an object may prevent addition of a further object.

Once objects are in the queue, the DLS provider may manipulate pairs of adjacent objects, resulting in deletion. An object may be deleted if, and only if, the object which follows it is defined to be destructive with respect to the object. If necessary the last object in the queue will be deleted to allow a destructive object to be entered - they may therefore always be added to the queue. Disconnect objects are defined to be destructive with respect to all other objects. Reset objects are defined to be destructive with respect to all other objects except connect, disconnect objects.

The relationships between objects which may be manipulated in the above fashion are summarized in Table 1/X.212.

Whether the DLS provider performs actions resulting in deletion or not will depend upon the behaviour of the DLC users and the agreed QOS for the DLC. In general, if a DLS user does not remove objects from a queue, the DLS provider shall, after some unspecified period of time, perform all the permitted deletions.

TABLE 1/X.212

**Relationships between queue model objects**

| The following y is defined with respect to the preceding object x | Connect | Data | Reset | Synchronization mark | Disconnect |
|---|---|---|---|---|---|
| Connect | N/A | - | - | N/A | DES |
| Data | N/A | - | DES | N/A | DES |
| Reset | N/A | - | DES | - | DES |
| Synchronization mark | N/A | - | DES | N/A | DES |
| Disconnect | N/A | N/A | N/A | N/A | DES |

N/A: x will not precede y in a valid state of a queue

- : Not to be destructive nor to be able to advance ahead

DES: To be destructive to the preceding object

9.2.4 *Reset*

In order to accurately model the reset service a synchronization mark object is required. The synchronization mark object exhibits the following properties:

a) it cannot be removed from a queue by a DLS-user;

b) a queue appears empty to a DLS-user when a synchronization mark object is the next object in the queue;

c) a synchronization mark object can be destroyed by a Disconnect object (see Table 1/X.212);

d) when a Reset object is immediately preceded by a synchronization mark object, both the Reset object and the synchronization mark object are deleted from the queue.

The initiation of a reset procedure is represented in the two queues as follows:

i) initiation of a reset procedure by the DLS provider is represented by the introduction into each queue of a reset object followed by a synchronization mark object;

ii) a reset procedure initiated by a DLS user is represented by the addition, by the DLS provider, of a reset object into the queue from the Reset initiator to the peer DLS user and the insertion of a reset object followed by a synchronization mark object into the other queue.

Unless destroyed by a disconnect object, a synchronization mark object remains in the queue until the next objet following it in the queue is a reset object. Both the synchronization mark object and the following reset object are then deleted by the DLS provider.

*Note* - Associated with the initiation of a reset procedure are restrictions on the issuance of certain other types of primitives. These restrictions will result in restrictions on the entry of certain object types into the queue until the reset procedure is completed (see § 14.2.3).

9.2.5 *DLC release*

The insertion into a queue of a disconnect object, which may occur at any time, represents the initiation of a DLC release procedure. The release procedure may be destructive with respect to other objects in the two queue and eventually results in the emptying of the queues and the disassociation of the queues with the DLC.

The insertion of a disconnect object may also represent the rejection of a DLC establishment attempt or the failure to complete DLC establishment. In such cases, if a connect object representing a DL-CONNECT request primitive is deleted by a disconnect object, then the disconnect object is also deleted. The disconnect object is not deleted when it deletes any other object, including the case where it deletes a connect object representing a DL-CONNECT response.

## 10 Quality of connection-mode Data Link Service

The term "Quality of Service" (QOS) refers to certain characteristics of a DLC as observed between the connection endpoints. QOS describes aspects of a DLC that are attributable solely to the DLS provider.

Once a DLC is established, the DLS users at the two ends have the same knowledge and understanding of what the QOS over the DLC is.

10.1 *Determination of QOS for Connection-mode Service*

QOS is determined in terms of QOS parameters. These parameters give DLS users a method of specifying their needs and give the DLS provider a basis for protocol selection.

The DLS QOS parameters can be divided into the following two types, based upon the way in which their values are determined:

a) those QOS parameters which may be selected on a per-connection basis during the establishment phase of a DLC;

b) those QOS parameters which are not selected during DLC establishment but whose values are known by other methods.

There are three QOS parameters, throughput, protection, and priority (as defined in §§ 10.2.1, 10.2.5 and 10.2.6 respectively), which are of the type that may be selected during DLC establishment. The selection procedures for these parameters are described in detail in § 12.2.5. Once the DLC is established, throughout the lifetime of the DLC, the agreed QOS values are not reselected at any point, and there is no guarantee that the original values will be maintained. The DLS users should also be aware that changes in QOS on a DLC are not explicitly signalled by the DLS provider.

The remaining QOS characteristics that are identified as parameters, but for which there is no selection during DLC establishment, are defined in §§ 10.2.2 through 10.2.4 respectively. The values of these parameters for a particular DLC are determined by other methods such as ***a*** priori knowledge and agreement.

If selection is allowed, certain measures of QOS are requested by the sending DLS user when the DL-CONNECT request primitive action is initiated. The requested measures (or parameter values and options) are based on *a priori* knowledge by the DLS user of the service(s) made available to it by the DLS provider. Knowledge of the characteristics and type of service provided (i.e. the parameters, formats, and options that affect the transfer of data) is made available to a DLS user through some layer management interaction prior to (any) invocation of the DL connection-mode service. Thus, the DLS user has explicit knowledge of the characteristics of the service it can expect to be provided with each invocation of the service.

The DLS provider may also provide information on the current QOS independently of access to the service by the DLS user. This seemingly dynamic aspect of QOS determination is not a negotiation but provided with knowledge of the characteristics of the service currently outside of any instance of the invocation of the service.

10.2 *Definition of connection-mode QOS parameters* QOS parameters can be classified as:

a) Parameters that express DLS performance, as shown in Table 2/X.212.

TABLE 2/X.212

**Classification of performance QOS parameters**

| Performance criterion | |
|---|---|
| Speed | Accuracy reliability |
| Throughput transit delay | Residual error rate (corruption, duplication/loss)<br><br>Resilience |

b) parameters that express other DLS characteristics, as shown in Table 3/X.212.

TABLE 3/X.212

**QOS parameters not associated with performance**

| |
|---|
| Protection |
| Priority |

*Note* - Some QOS parameters are defined in terms of the issuance of DLS primitives. Reference to a DLS primitive refers to the complete execution of that DLS primitive at the appropriate DLSAP.

10.2.1 *Throughput*

Throughput is defined as the total number of DLSDU bits successfully transferred by a DL-DATA request/DL-DATA indication primitive sequence divided by the input/output time for that sequence.

Successful transfer of the bits in a transmitted DLSDU is defined to occur when the bits are delivered to the intended receiving DLS user without error, in the proper sequence, prior to release of the DLC by the receiving DLS user.

The input/output time for a DL-DATA request/DL-DATA indication primitive sequence is the greater of the two times in the following list:

a) the time between the first and the last DL-DATA request in the sequence;

b) the time between the first and the last DL-DATA indication in the sequence.

Throughput is only meaningful for a sequence of complete DLSDUs.

Throughput is specified independently for each direction of transfer. In general, each throughput specification will define both the desired target value and the minimum acceptable value (or lowest acceptable QOS) for a DLC. Each specification is an average rate and will be based on a previously stated average DLSDU size.

Either the input or the output of a sequence of DLSDUs may be delayed excessively by the DLS users. Occurrences of delay caused by the DLS users are excluded in calculating average throughput values.

10.2.2 *Transit delay*

Transit delay is the elapsed time between DL-DATA request primitives and the corresponding DL-DATA indication primitives. Elapsed time values are calculated only on DLSDUs that are successfully transferred.

Successful transfer of a DLSDU for the purposes of the QOS parameter is defined to occur when the DLSDU is transferred from the sending DLS user to the intended receiving DLS user without error, and in the proper sequence prior to release of the DLC by the receiving DLS user.

For connection-mode transfer transit delay is specified independently for each direction of transfer. Each specification is based on a previously stated average DLSDU size.

The transit delay for an individual DLSDU may be increased if the receiving DLS user exercises interface flow control. Such occurrences are excluded in calculating transit delay values.

10.2.3 *Residual Error Rate*

Residual error rate is the ratio of total incorrect, lost and duplicate DLSDUs to total DLSDUs transferred across the DLS boundary during a measurement period. The relationship among these quantities is defined, for a particular DLS user pair, as shown in Figure 3/X.212.

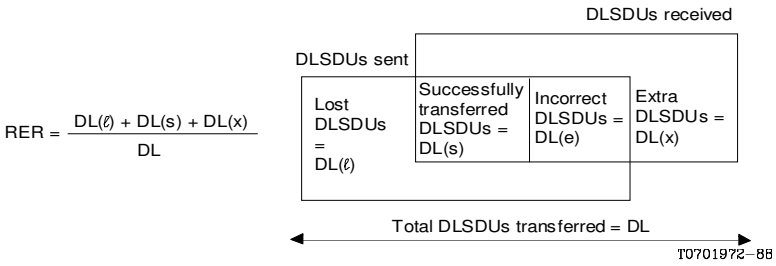


FIGURE 3/X.212

**Components of Residual Error Rate**

10.2.4 *Resilience*

This parameter specifies the probability of:

a) a DLS provider initiated DLC release (i.e. the issuance of a DL-DISCONNECT indication primitive with no prior DL-DISCONECT request primitive); or

b) a DLS provider initiated reset (i.e. the issuance of a DL-RESET indication primitive with no prior DL-RESET request primitive);

during a specified time interval on an established DLC.

10.2.5 *Protection*

Protection is the extent to which a DLS provider attempts to prevent unauthorized monitoring or manipulation of DLS user originated information. Protection is specified by a minimum and maximum protection option within a range of three possible protection options:

a) no protection features;

b) protection against passive monitoring; and

c) protection against modification, replay, addition or deletion.

Within the specified range, a DLS user selects a particular value during DLC establishment.

Each protection feature addresses a particular type of privacy or security threat, and each if available is typically provided by a different DLS provider mechanism.

10.2.6 *Priority*

The specification of priority is concerned with the relationship between DLCs.

This parameter specifies the relative importance of a DLC with respect to:

a) the order in which DLCs are to have their QOS degraded, if necessary; and

b) the order in which DLCs are to be released to recover resources, if necessary.

Priority is specified by a minimum and a maximum within a given range. Within the specified range, a DLS user selects a particular value during DLC establishment.

This parameter only has meaning in the context of some management entity of structure able to judge relative importance. The number of priority levels is limited.

## 11 Sequence of primitives

11.1 *Concepts used to define the connection-mode Data Link Service*

The service definition uses the following concepts:

a) DLC can be dynamically established or terminated between the DLS users for the purpose of exchanging data;

b) associated with each DLC, certain measures of QOS that are agreed between the DLS provider and the DLS users when the connection is established;

c) the DLC allows transmission of data and preserves its division into DLSDUs; the transmission of this data is subject to flow control;

d) the DLC can be returned to a defined state, and the activities of the two DLS users synchronized by the use of a Reset Service;

e) failure to provide the requested service may be signalled to the DLS user. There are three classes of failure:

   1) failures involving termination of the DLC;

   2) failures involving loss or duplication of user data, but without loss of DLC; and

   3) failures to provide the requested QOS without loss or duplication of user data or loss of the DLC.

11.2 *Constraints of Sequence of Primitives*

This section defines the constraints of the sequence in which the primitives defined in sections 12-14 may occur. The constraints determine the order in which primitives occur, but do not fully specify when they may occur. Other constraints, such as flow control of data, will affect the ability of a DLS user or a DLS provider to issue a primitive at any particular time.

The connection-mode primitives and their parameters are summarized in Table 4/X.212.

TABLE 4/X.212

**Summary of data-link-connection-mode primitives and parameters**

| Phase | Service | Primitive | Parameters |
|---|---|---|---|
| DLC establishment | DLC establishment | DL-CONNECT request | (Called address, calling address, QOS parameter set) |
| | | DL-CONNECT indication | (Called address, calling address, QOS parameter) |
| | | DL-CONNECT response | (Responding address, QOS parameter set) |
| | | DL-CONNECT confirm | (Responding address, QOS parameter set) |
| Data transfer | Normal data transfer | DL-DATA request | (DLS user-data) |
| | | DL-DATA indication | (DLS user-data) |
| | Reset | DL-RESET request | (Reason) |
| | | DL-RESET indication | (Originator, reason) |
| | | DL-RESET response | |
| | | DL-RESET confirm | |
| DLC release | DLC release | DL-DISCONNECT request | (Reason) |
| | | DL-DISCONNECT indication | (Originator reason) |

11.2.1 *Relation of Primitives at the Two Endpoints*

A primitive issued at one DLC endoint will, in general, have consequences at the other DLC endpoint. The relations of primitives of each type at one DLC endpoint to primitives at the other DLC endpoint are defined in the appropriate subsections in sections 12-15; all these relations are summarized in the diagrams in Figure 4/X.212.

However, a DL-DISCONNECT request or indication primitive may terminate any of the other sequences before completion.

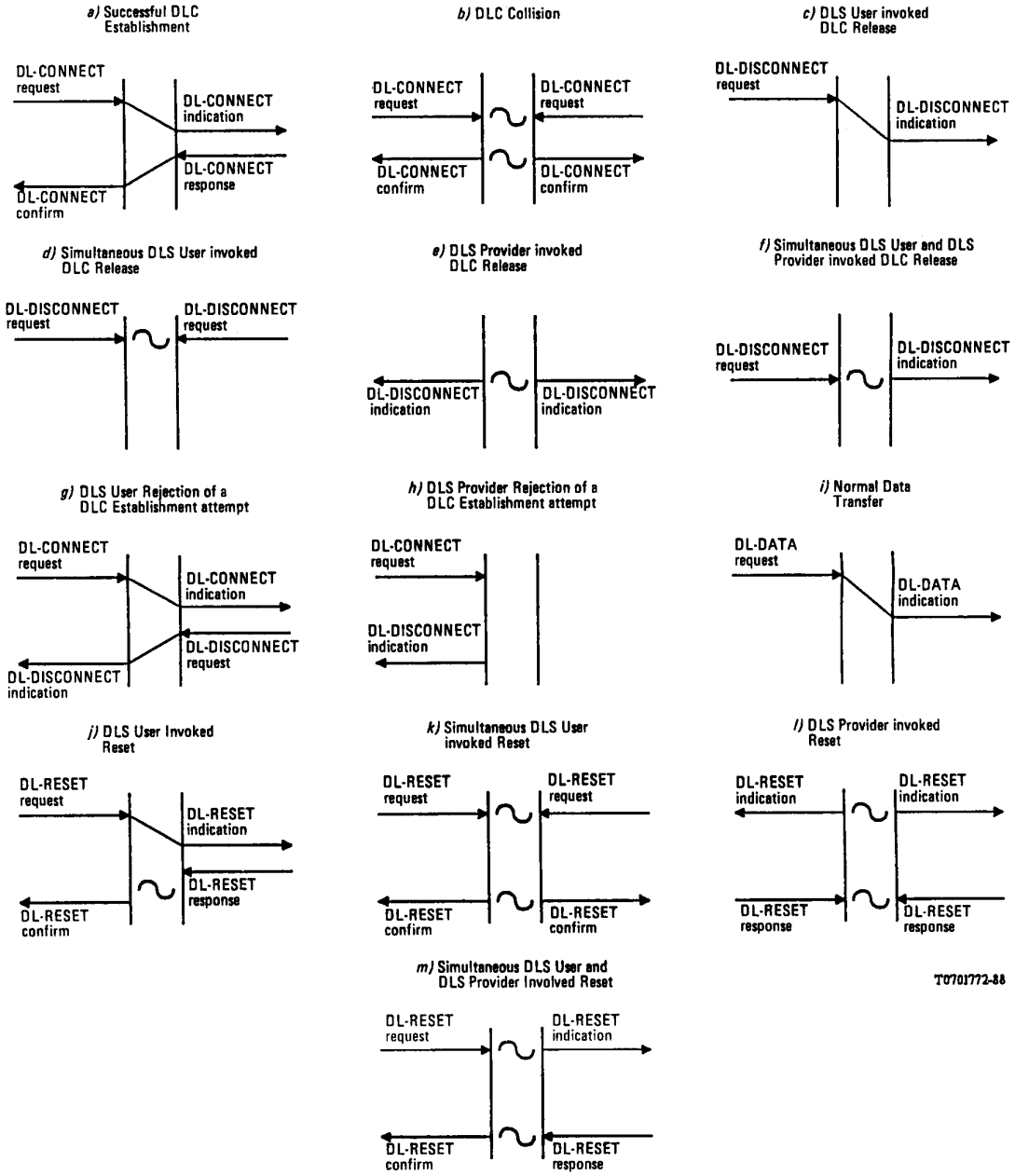


FIGURE 4/X.212

**Summary of Data-link-connection-mode Service Primitive Time-sequence Diagram**

11.2.2 *Sequence of primitives al one DLC endpoint*

The possible overall sequences of primitives at a DLC endoint are defined in the state transition diagram, Figure 5/X.212. In the diagram:

a) DL-DISCONNECT stands for either the request or the indication form of the primitive in all cases.

b) The labelling of the states "DLS user initiated reset pending" (5) and "DLS provider initiated reset pending" (6) indicate the party that started the local interaction, and does not necessarily reflect the value of the originator parameter.

c) The idle state (1) reflects the absence of a DLC. It is the initial and final state of any sequence, and once it has been re-entered, the DLC 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 implementations of the service.

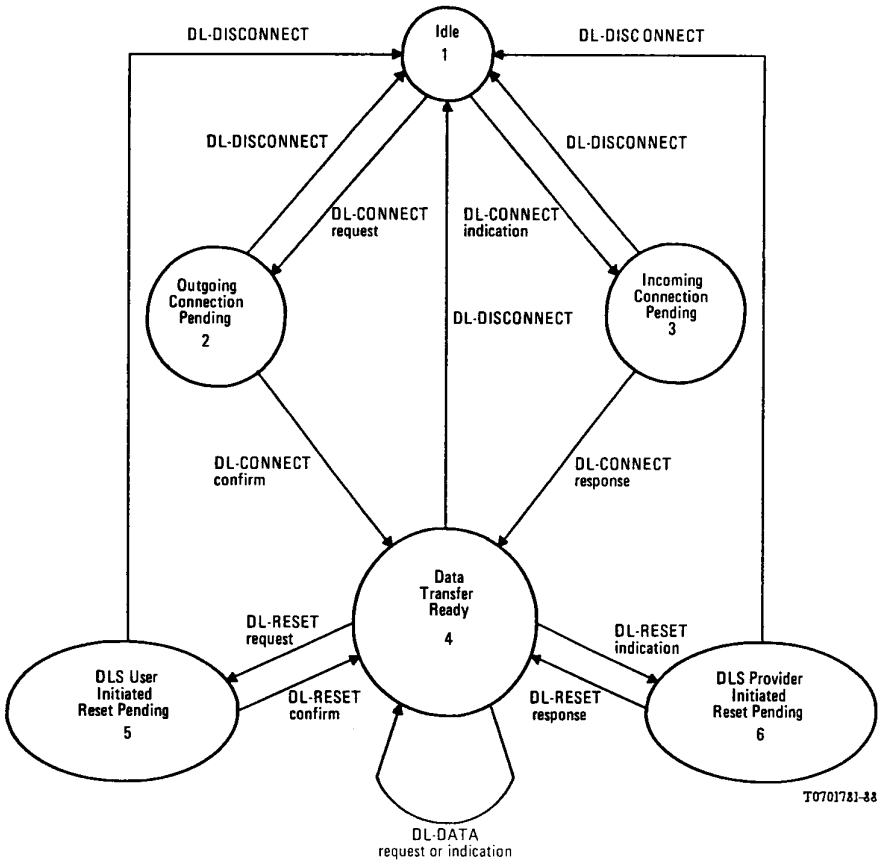


FIGURE 5/X.212

**State transition diagram for Sequences of Data-link-connection-mode Service primitives at a DLC End-point**

## 12 Connection establishment phase

12.1 *Function*

The connection establishment service primitives can be used to establish a DLC.

Simultaneous DL-CONNECT request primitives at the two DLSAPs result in one DLC as indicated in Figure 6/X.212.

12.2 *Types of primitives and parameters*

Table 5/X.212 indicates the types of primitives and the parameters needed for connection establishment.

TABLE 5/X.212

**DLC establishment primitives and parameters**

| Primitive / Parameter | DL-CONNECT request | DL-CONNECT indication | DL-CONNECT response | DL-CONNECT confirm |
|---|---|---|---|---|
| Called address | X | X(=) (see Note 2) | | |
| Calling address | X (see Note 2) | X(=) | | |
| Responding address | | | X (see Notes 1, 2) | X(=) |
| Quality of service parameter set | X | X | X | X |

*Note 1* - The need for responding address parameter is for further study.

*Note 2* - This parameter may be implicitly associated with the DLSAP at which the primitive is issued.

12.2.1 *Addresses*

The parameters which take addresses as values (see §§ 12.2.2-12.2.4) all refer to DLSAP addresses.

*Note* - If the configuration allows any of these addresses to be known by the DL Entity on an *a priori* basis, then these DLSAP address(es) need not explicitly be conveyed in the protocol.

12.2.2 *Called Address*

The called address parameter conveys an address identifying the DLSAP to which the DLC is to be established.

12.2.3 *Calling Address*

The calling address parameter conveys the address of the DLSAP from which the DLC has been requested.

12.2.4 *Responding Address*

The responding address parameter conveys the address of the DLSAP to which the DLC has been established.

12.2.5 *Quality of Service parameter set*

The use of the QOS parameter selection is not required when only one level of QOS is offered by the DLS provider.

12.2.5.1 *Throughput*

Two quality of service sub-parameters "target" and "lowest quality acceptable", which are in the agreed range, are passed to the DLS provider in the CONNECT request primitive. THE DLS provider will indicate to the DLS users, the "available" throughput in the DL-CONNECT confirm, and DL-CONNECT indication. The "available" parameter shall be a value from the range between the "target" and "lowest quality acceptable" (see § 10.2.1).

12.2.5.2 *Selected protection*

The parameter specifies a particular degree of protection, within the agreed range (see § 10.2.5) for the DLSDU of any subsequently submitted DL-DATA request primitive transferred on the DLC.

12.2.5.3 *Selected priority*

The parameter specifies a particular priority, within the agreed range (see § 10.2.6), for the DLSDU of any subsequent DL-DATA request primitive transferred on the DLC.

12.3 *Sequence of primitives*

The sequence of primitives in a successful connection establishment is defined by the time-sequence diagram in Figure 6/X.212.

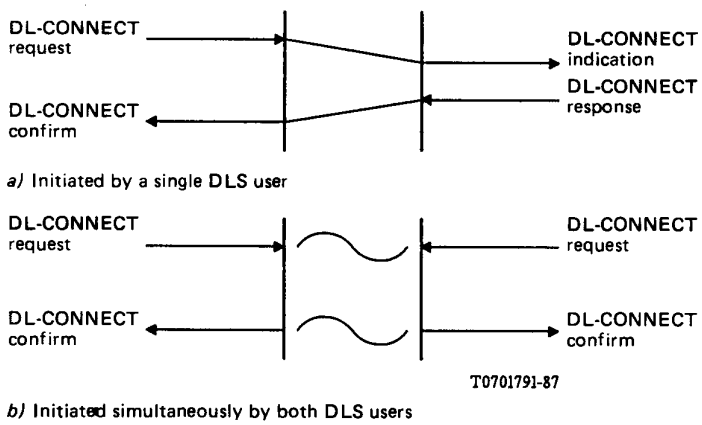


FIGURE 6/X.212

**DLC Establishment**

These DLC establishment procedures may fail either due to the inability of the DLS provider to establish a DLC or due to the unwillingness of the called DLS user to accept a DL-CONNECT indication primitive (for these cases see DLC release service, §§ 13.4 and 13.5).

## 13 Connection release phase

13.1 *Function*

The connection release service primitives are used to release a DLC. The release may be initiated by any of the following:

a) either or both of the DLS users, to release an established DLC;

b) the DLS providers to release an established DLC; all failures to maintain a DLC are indicated in this way;

c) the DLS user, to reject a DL-CONNECT indication;

d) the DLS provider, to indicate its inability to establish a requested DLC; or

e) the DLS user which sent the DL-CONNECT request, to abandon the connection attempt before the connection has been made available for use by receipt of a DL-CONNECT primitive.

Initiation of the release service element is permitted at any time regardless of the current phase of the DLC. Once a release service has been initiated, the DLC will be disconnected. A DL-DISCONNECT request cannot be rejected. The DLS does not guarantee delivery of any DLSDU associated with the DLC once the release phase is entered.

13.2 *Types of primitive and parameters*

Table 6/X.212 indicated the types of primitives and the parameters needed for connection release.

TABLE 6/X.212

**DLC release primitives and parameters**

| Primitive / Parameter | DL-DISCONNECT request | DL-DISCONNECT indication |
|---|---|---|
| Originator | | X |
| Reason | X | |

13.2.1 *Originator*

The originator parameter indicates the source of the release. Its value indicates either the DLS user, the DLS provider, or that the originator is unknown.

13.2.2 *Reason*

The reason parameter gives information about the cause of the release. The value conveyed in this parameter will be as follows:

a) when the originator parameter indicates a DLS provider generated release, the value is one of:

1) "disconnection-permanent condition";

2) "disconnection-transient condition";

3) "connection rejection-DLSAP-address unknown";

4) "connection rejection-DLSAP unreachable/permanent condition";

5) "connection rejection-DLSAP unreachable/transient condition";

6) "connection rejection-QOS not available/permanent condition";

7) "connection rejection-QOS not available/transient condition"; or

8) "reason unspecified";

*Note* - Addition to, or refinement of this list of values to convey more specific diagnostic and management information is for further study.

b) when the originator parameter indicates DLS user initiated release, the value is one of:

1) "disconnection-normal condition";

2) "disconnection-abnormal condition";

3) "connection rejection-permanent condition";

4) "connection rejection-transient condition"; or

5) "reason unspecified"; and

c) when the originator parameter indicates an unknown originator the value of the Reason parameter is "reason unspecified". This allows the parameters to be implied when they cannot be explicitly conveyed in the Data Link protocol.

13.3 *Sequence of primitives when releasing an established DLC*

The sequence of primitives depends on the origins of the release action. The sequence may be:

a) initiated by one DLS user, with a request from that DLS user leading to an indication to the other;

b) initiated by both DLS users, with a request from each of the DLS users;

c) initiated by the DLS provider, with an indication to each of the DLS users; or

d) initiated independently by one DLS user and the DLS provider, with a request from the originating DLS user and an indication to the other.

The sequences of primitives in these four cases are defined by the time-sequence diagrams in Figures 7/X.212-10/X.212.

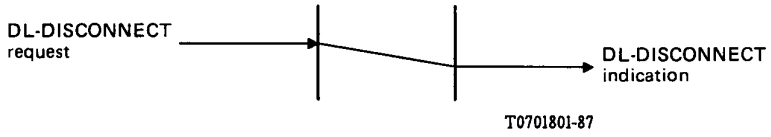


FIGURE 7/X.212

**DLS User invocation**

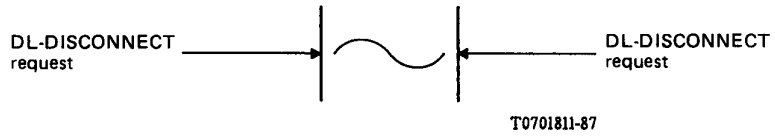


FIGURE 8/X.212

**Simultaneous invocation by both DLS Users**

FIGURE 9/X.212

**DLS Provider invocation**

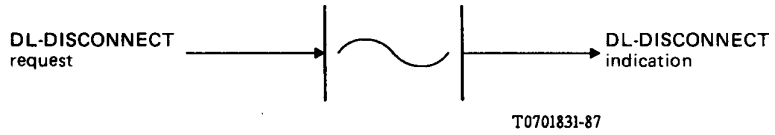


FIGURE 10/X.212

**Simultaneous DLS User and DLS Provider invocations**

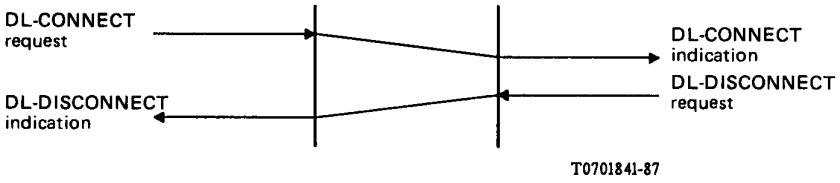


FIGURE 11/X.212

**Sequence of primitives in DLS User rejection of a DLC connection attempt**

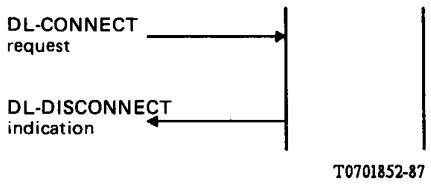


FIGURE 12/X.212

**Sequence of primitive in DLS Provider rejection of a DLC establishment attempt**

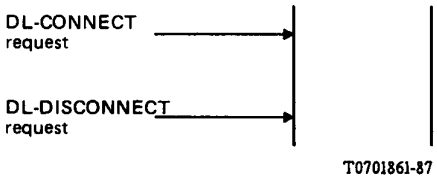


FIGURE 13/X.212

**Both primitives are destroyed in the queue**

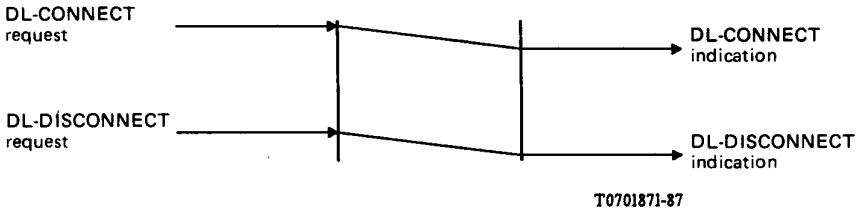


FIGURE 14/X.212

**DL-DISCONNECT indication arrives before DL-CONNECT response is sent**

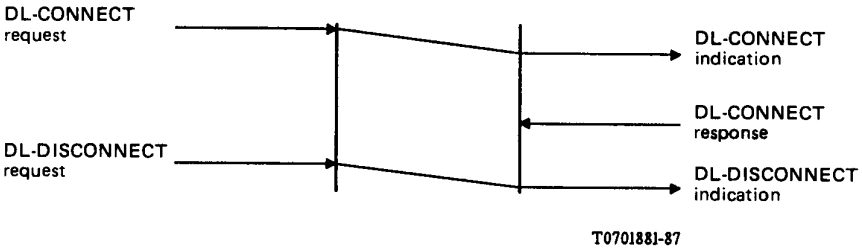


FIGURE 15/X.212

**DL-DISCONNECT indication arrives after DL-CONNECT response is sent**

13.4 *Sequence of primitives in a DLS user rejection of DLC establishment attempt*

A DLS user may reject a DLC establishment attempt by using a DL-DISCONNECT request. The originator parameter in the DL-DISCONNECT primitives will indicate DLS user initiated release. The sequence of events is defined in the time-sequence diagram in Figure 11 /X.212.

13.5 *Sequence of primitives in a DLS provider rejection of a DLC establishment attempt*

If the DLS provider is unable to establish a DLC, it indicates this to the requester by a DL-DISCONNECT indication. The originator parameter in this primitive indicates a DLS provider originated release. The sequence of events is defined in the time-sequence diagram in Figure 12/X.212.

13.6 *Sequence of Primitives in a DLS User Abort of a DLC Establishment Attempt*

If the DLS user, having previously sent a DL-CONNECT request and not received a DL-CONNECT confirm or DL-DISCONNECT indication, wishes to abort the DLC establishment attempts the DLS user shall issue a DL-DISCONNECT request. The resulting sequence of primitives is dependent upon the relative timing of the primitives involved and the transit delay of the DLS provider as defined by the time-sequence diagrams in Figures 13 to 15/X.212. No information can be implied by detecting which of these alternatives occur.

## 14 Data transfer phase

14.1 *Data transfer*

14.1.1 *Function*

The data transfer service primitives provide for an exchange of user-data (DLSDUs), in either direction or in both directions simultaneously on a DLC. The DLS preserves both the sequence and the boundaries of the DLSDUs.

*Note* - Designers of protocols using DLS should realize that the requested QOS applies to complete DLSDUs, and that divisions of available data into small DLSDUs may have cost implications because of the impact on cost optimization mechanisms operated by the DLS provider.

14.1.2 *Types of primitives and parameter*

Table 7/X.212 indicates the types of primitives and the parameters needed for data transfer.

TABLE 7/X.212

**Data transfer primitives and parameter**

| Primitive / Parameter | DL-DATA request | DL-DATA indication |
|---|---|---|
| DLS user-data | X | X(=) |

14.1.2.1 *DLS user-data*

This parameter allows the transmission of DLS user-data between DLS users, without modification by the DLS provider. The DLS user may transmit any integral number of octets greater than zero up to a limit determined by the DLS provider. The value of this limit is made available to the DLS user by the use of management facilities or *a priori* knowledge.

14.1.3 *Sequence of primitives*

The operation of the DLS in transferring DLSDUs can be modelled as a queue of unknown size within the DLS provider (see § 9). The ability of a DLS user to issue a DL-DATA request or of the DLS provider to issue a DL-DATA indication depends on the behaviour of the receiving DLS user and the resulting state of the queue.

The sequence of primitives in a successful data transfer is defined in the time-sequence diagram in Figure 16/X.212.

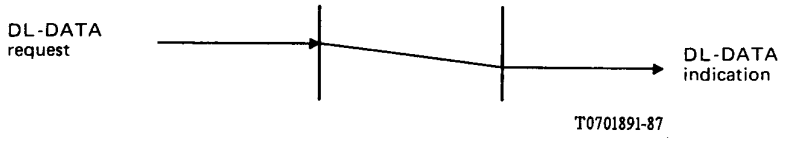


FIGURE 16/X.212

**Sequence of primitives for normal data transfer service**

The above sequence of primitives may remain uncompleted if a DL-RESET or a DL-DISCONNECT primitive occurs.

14.2 *Reset service*

14.2.1 *Function*

The Reset service may be used:

a) by the DLS user, to resynchronize the use of the DLC; or

b) by the DLS provider, to report detected loss of data unrecoverable within the DLS. All loss of data which does not involve loss of the DLC is reported in this way.

Invocation of the Reset service will unblock the flow of DLSDUs in case of congestion of the DLC; it will cause the DLS provider to discard DLSDUs, and to notify user or users that did not invoke Reset that a Reset has occurred. The service will be completed in a finite time, irrespective of the acceptance of DLSDUs. Any DLSDUs not delivered to the DLS users before completion of the service will be discarded by the DLS provider.

*Note* - A Reset may require a recovery procedure to be performed by the DLS users.

14.2.2 *Types of primitives and parameters*

Table 8/X.212 indicates the types of primitives and the parameters needed for the reset service.

TABLE 8/X.212

**Reset primitives and parameters**

| Primitive / Parameter | DL-RESET request | DL-RESET indication | DL-RESET response | DL-RESET confirm |
|---|---|---|---|---|
| Originator | | X | | |
| Reason | X | X | | |

14.2.2.1 *Originator*

The originator parameter indicates the source of the Reset. Its value indicates either the DLS user, the DLS provider, or that the originator is unknown.

14.2.2.2 *Reason*

The reason parameters give information indicating the cause of the Reset. The value conveyed in this parameter will be as follows:

a) when the originator parameter indicates a DLS provider generated Reset, the value is one of:

1) "Data Link flow control congestion"; or

2) "Data Link error";

*Note* - Addition to or refinement of this list of values to convey more specific diagnostic or management information is for further study.

b) when the originator parameter indicates a DLS user initiated Reset, the value is "user resynchronization"; and

c) when the originator parameter indicates an unknown originator, the value is "reason unspecified". This allows the parameters to be implied when they cannot be explicitly conveyed in the Data Link protocol.

14.2.3 *Sequence of primitives*

The interaction between each DLS user and the DLS provider shall be either one of the following exchanges of primitives:

a) a DL-RESET request from the DLS user, followed by a DL-RESET confirm from the DLS provider; or

b) a DL-RESET indication from the DLS provider, followed by a DL-RESET response from the DLS user.

The DL-RESET request acts as a synchronization mark in the stream of DLSDUs that are transmitted by the issuing DLS user; the DL-RESET indication likewise acts as a synchronization mark in the stream of DLSDUs by the peer DLS user. Similarly, the DL-RESET response acts as a synchronizing mark in the stream of DLSDUs transmitted by the responding DLS user, while the DL-RESET confirmation acts as a synchronization mark in the stream of DLSDUs that are received by the DLS user which originally issued the reset.

The resynchronization properties of the Reset Service are that:

1) No DLSDU transmitted by the DLS user *before* the synchronization mark in that transmitted stream will be delivered to the other DLS user *after* the synchronization mark in that received stream.

   The DLS provider will discard all DLSDUs, submitted before the issuing of the DL-RESET request that have not been delivered to the peer DLS user when the DLS provider issues the DL-RESET indication.

   Also, the DLS provider will discard all DLSDUs, submitted before the issuing of the DL-RESET that have not been delivered to the initiator of the DL-RESET when the DLS provider issues the DL-RESET confirm.

2) No DLSDU transmitted by a DLS user *after* the synchronization mark in that transmitted stream will be delivered to the other DLS user *before* the synchronization mark in that received stream.

The complete sequence of primitives depends upon the origin of the Reset action and the occurrence or otherwise of conflicting origins. Thus the Reset Service may be:

i) invoked by one DLS user, leading to interaction a) with that DLS user and interaction b) with the peer DLS user;

ii) invoked by both DLS users, leading to interaction a) with both DLS users;

iii) invoked by the DLS provider, leading to interaction b) with both DLS users; or

iv) invoked by one DLS user and DLS provider, leading to interaction a) with the originating DLS user and b) with the peer DLS user.

The sequence of primitives in these four cases is defined in the time-sequence diagrams in Figures 17-20/X.212.

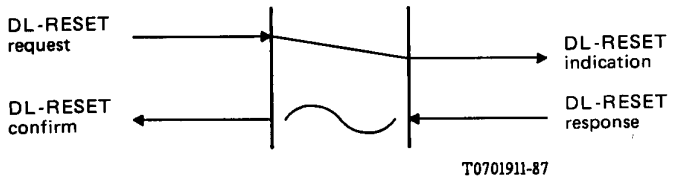


FIGURE 17/X.212
**Sequence of primitives in a DLS user initiated reset service**

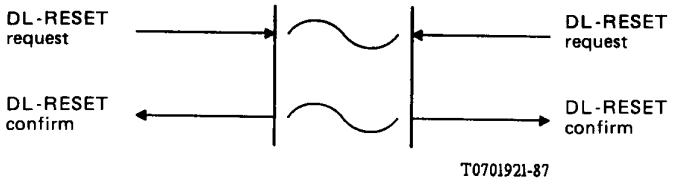


FIGURE 18/X.212

**Sequence of primitives in a simultaneous DLS user initiated reset service**

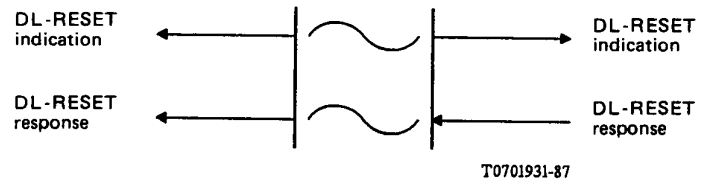


FIGURE 19/X.212
**Sequence of primitives in a DLS provider initiated reset service**

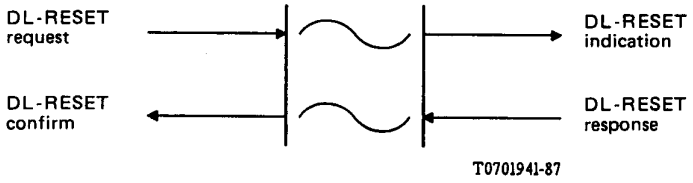


FIGURE 20/X.212
**Sequence of primitives in a simultaneous DLS user and DLS provider initiated reset service**

The above sequences of primitives may remain uncompleted if a DL-DISCONNECT primitive occurs.

## PART THREE - DEFINITION OF THE CONNECTIONLESS-MODE SERVICE

### 15 Features of the Connectionless-mode Data Link Service

The DLS provides the following features to the DLS user:

a) a means by which DLSDUs of limited length are delimited and transparently transmitted from one source DLSAP to a destination DLSAP in a single DLS access, without establishing or later releasing a DLC;

b) associated with each instance of connectionless-mode transmission, certain measures of QOS which are selected by the sending DLS user when the connectionless-mode transmission is initiated.

### 16 Model of the Connectionless-mode Data Link Service

This Recommendation uses the abstract model for a layer service defined in § 4 of Recommendation X.210. The model defines the interactions between the DLS users and DLS provider which takes place at the two DLSAPs. Information is passed between the DLS user and the DLS provider by service primitives, which may convey parameters.

16.1 *Model of a Data-link-connectionless-mode Data Transmission*

A defining characteristic of data-link-connectionless-mode data transmission is the independent nature of each invocation of the DL-connectionless-mode service (see Appendix 11).

In practice, however, it is often possible to relate to DLS users certain characteristics of the service for an association existing between a given pair of DLSAPs, which enhance the basic data-link-connectionless-mode service in order to effectively correlate the choice of Network Layer protocol type with the service provided.

*Note - It* is anticipated that such information is made available to the DLS user through some management facility (or set of facilities).

Thus as a descriptive aid the data-link-connectionless-mode service - as provided between any two DLSAPs - can be modelled in the abstract as an association between the two DLSAPs. This association is permanent.

Only one type of object, the unitdata object, can be handed over to the DLS provider via a DLSAP. In Figure 21/X.212, DLS user A represents the DLS user that passes objects to the DLS provider. DLS user B represents the DLS user that accepts objects from the DLS provider.

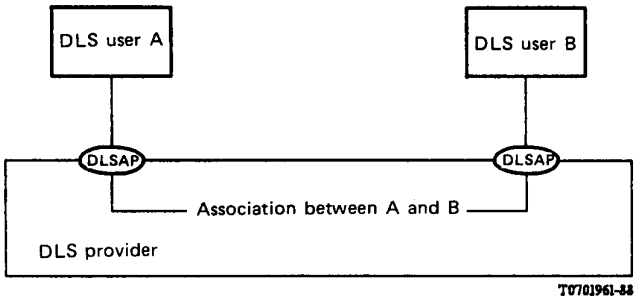


FIGURE 21/X.212

**Model for a connectionless DLS data transmission**

In general, the DLS provider may perform any or all of the following actions:

a) discard objects;

b) duplicate objects; and/or

c) change the order of service request into a different order of service indications.

However, with respect to a given association, some characteristics of the nature and type of service, beyond those attributed to the basic DL-connectionless-mode service, may be related to the DLS user through some management facility. The following are examples of some requirements or constraints that may be assumed/observed by the DLS user:

a) objects will not be discarded;

b) objects will not be duplicated; and

c) the order of the service indications will be the same as the order of the service requests.

Where such information is made known to the DLS user prior to the invocation of the DL-connectionless-mode service, it may make use of such knowledge to select an appropriate Network Layer protocol.

The operations that are performed by the DLS provider for a particular DL association do not depend on the behaviour of the DLS users. Awareness of the characteristics of the DLS provided is part of the DLS users' *a priori* knowledge of the OSI environment.

## 17 Quality of Connectionless-mode Service

The term "Quality of Service" (QOS) refers to certain characteristics of a connectionless-mode data transmission as observed between the DLSAPs. QOS describes aspects of a connectionless-mode data transmission which are solely attributable to the DLS provider; it can only be properly determined in the absence of DLS user behaviour (which is beyond the control of the DLS provider) that specifically constrains or impedes the performance of the DLS.

Whether the view of the QOS during each instance of the use of connectionless-mode data transmission is the same to each DLS user associated with the service depends on the nature of their association and the type of information concerning the nature of the service made available to the DLS user(s) by the DLS provider prior to the invocation of the service.

### 17.1 *Determination of QOS for Connectionless-mode Service*

A basic characteristic of a connectionless-mode service is that, unlike a connection-mode service, no dynamic association similar to that during a connection establishment is set up between the parties involved. Thus, the service characteristics to be provided during the transfer are not selected on a per DLC basis.

Associated with each DL connectionless-mode transmission, certain measures of QOS are requested by the sending DLS user when the primitive action is initiated. The requested measures (or parameter values) and options, are based on *a priori* knowledge by the DLS user of the service(s) made available to it by the DLS provider. Knowledge of the characteristics and type of service provided (i.e., the parameters, formats, and options that affect the transfer of data) is made available to a DLS user through some layer management interaction prior to (any) invocation of the DL-connectionless-mode service. Thus, the DLS user not only has knowledge of the parties with which it may communicate, it also has explicit knowledge of the characteristics of the service it can expect to be provided with each invocation of the service.

The DLS provider may also provide information on the current QOS independently of access to the service by a DLS user. This seemingly dynamic aspect of QOS determination is not a negotiation but provided with knowledge of the characteristics of the service currently outside of any instance of the invocation of the service.

17.2 *Definition of connectionless-mode QOS parameters*

QOS parameters are classified as:

a) parameters that express DLS performance, as shown in Table 9/X.212.

TABLE 9/X.212

**Classification of performance QOS-parameters**

| Performance criterion | |
|---|---|
| Speed | Accuracy/reliability |
| Transit delay | Residual error rate (corruption, duplication/loss) |

b) parameters that express other DLS characteristics, as shown in Table 1 0/X.212.

TABLE 10/X.212

**QOS-parameters not associated with performance**

| |
|---|
| Protection |
| Priority |

*Note* - Some QOS-parameters are defined in terms of the issuance of DLS primitives. Reference to a DLS primitive refers to the complete execution of that DLS primitive at the appropriate DLSAP.

17.2.1 *Transit delay*

Transit delay is the elapsed time between DL-UNITDATA request primitives and the corresponding DL-UNITDATA indication primitives. Elapsed time values are calculated only on DLSDUs that are successfully transferred.

Successful transfer of a DLSDU is defined, for the purpose of this QOS parameter, to occur when the DLSDU is transferred from the sending DLS user to the intended receiving DLS user without error.

For connectionless-mode transfer, transit delay is specified independently for each Data-link-connectionless-mode data transmission.

The transit delay for an individual DLSDU may be increased if the receiving DLS user exercises interface flow control. Such occurrences are excluded in calculating both average and maximum transit delay values.

17.2.2 *Residual Error Rate*

Residual error rate is the ratio of total incorrect, lost and duplicate DLSDUs to total DLSDUs transferred across the DLS boundary during a measurement period. The relationship among these quantities is defined, for a particular DLS user pair, as shown in Figure 22/X.212.

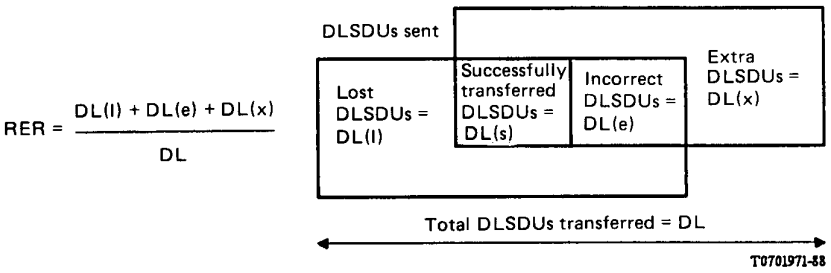


FIGURE 22/X.212

**Components of Residual Error Rate**

17.2.3 *Protection*

Protection is the extent to which a DLS provider attempts to prevent unauthorized monitoring or manipulation of DLS user originated information. Protection is specified by a minimum and maximum protection option within a range of three possible protection options:

a) no protection features;

b) protection against passive monitoring; and

c) protection against modification, replay, addition or deletion.

Within the specified range, a DLS user selects a particular value for each DLSDU submitted or connectionless-mode data transmission.

Each protection feature addresses a particular type of privacy or security threat and each is typically provided by a different DLS provider mechanism.

17.2.4 *Priority*

The specification of priority is concerned with the relationship between connectionless-mode data transfer invocations.

This parameter specifies the relative importance of unitdata objects with respect to gaining use of shared resources.

This parameter only has meaning in the context of some management entity of structure able to judge relative importance. The number of priority levels is limited.

## 18 Sequence of connectionless-mode primitives at one DLSAP

The possible overall allowed sequence of primitives at a DLSAP are defined in the state transition diagram in Figure 23/X.212.

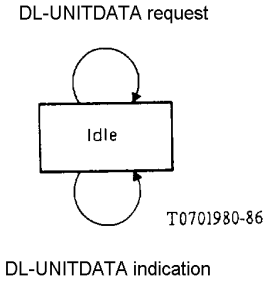


FIGURE 23/X.212

**State transition diagram for sequences of connectionless-mode primitives at one DLSAP**

## 19 Data transfer

19.1 *Function*

Data-Link connectionless-mode data transmission service primitives can be used to transmit an independent, self-contained DLSDU from one DLSAP to another DLSAP in a single DL service access. The DLSDU is independent in the sense that it bears no relationship to any other DLSDU transmitted through the invocation of the connectionless-mode service or the connection-mode service (unless specific QOS requests have been accepted). It is self-contained in that all of the information required to deliver the DLSDU is presented to the DLS provider, together with the user data to be transmitted, in a single service access; thus, no initial establishment or subsequent release of a DLS is required, provided that the DLS users exist and are known to the DLS provider.

A DLSDU transmitted using DL connectionless-mode data transmission is not considered by the DLS provider to be related in any way to any other DLSDU. Although the DLS maintains the integrity of individual DLSDUs, it does not necessarily deliver them to the receiving DLS user in the order in which they are presented by the sending DLS user.

No means are provided by which the receiving DLS user may control the rate at which the sending DLS user may send DLSDU (peer-to-peer flow control). The DLS provider will not maintain any state information relative to any aspects of the flow of information between any specific combination of DLSAPs. Flow control exerted by the DLS provider upon the sending DLS user can only be described in terms of a specific interface.

19.2 *Types of primitives and parameters*

Table 11 /X.212 indicates the types of primitives and parameters needed for the Data-link-connectionless-mode data transmission service.

TABLE 11/X.212

**DL-connectionless-mode data transfer primitives and parameters**

| Primitive / Parameter | DL-UNITDATA request | DL-UNITDATA indication |
|---|---|---|
| Source address | X | X(=) |
| Destination address | X | X(=) |
| QOS-parameter set | X | X (see Note) |
| DLS user-data | X | X(=) |

*Note* - The need for QOS-parameters to be included in the DL-UNITDATA indication is for further study.

19.2.1 *Addresses*

The addresses referred to in Table 11/X.212 are DLSAP addresses. The connection-mode and connection-less-mode DLSs may both use the same DLSAP addresses.

*Note* - If the configuration allows any of these addresses to be known by the DL Entity on an *a priori* basis, then these DLSAP addresses need not explicitly be conveyed in this protocol.

19.2.2 *Quality of Service*

The value of the QOS parameter is a list of sub-parameters. For each parameter, the values on the two primitives are related so that:

a) on the request primitive, any defined value is allowed; and

b) on the indication primitive, the quality of service indicated is less than or equal to the value specified for the corresponding request primitive.

The use of the QOS parameter selection is not required when only one level of QOS is offered by the DLS provider.

19.2.3 *DLS User-Data*

This parameter allows the transmission of DLS user-data between DLS users, without modification by the DLS provider. The DLS user may transmit any integral number of octets greater than zero up to a limit determined by the DLS provider. The value of this limit is made available to the DLS user by the use of management facilities or *a priori* knowledge.

19.3 *Sequence of primitives*

The sequence of primitives in a successful DL connectionless-mode data transmission is defined in the time-sequence diagram in Figure 24/X.212.

FIGURE 24/X.212

**Sequence of primitives in connectionless-mode data transfer**

APPENDIX I
(to Recommendation X.212)

## Differences between CCITT and ISO texts

The following differences between this Recommendation and ISO 8886, Information Processing Systems Data Communications - Data Link Service Definition, should be noted.

1. Appendix II to this Recommendation, which describes the relationship between connection-mode and connectionless-mode operation, is not present in ISO 8886.

2. Appendix III to this Recommendation, which defines a method for providing the OSI connection-mode Data Link Service using X.25 LAPB compatible DTE procedures, is not present in ISO 8886.

APPENDIX II
(to Recommendation X.212)

**The relationship between the two types of Data Link Service**

This appendix does not form a part of this Recommendation.

II.1 *Introduction*

Recommendation X.200 describes the Reference Model of Open Systems Interconnection for CCITT Applications. It is the intention of Recommendation X.200 that the Reference Model should establish a framework for coordinating the development of existing and future Recommendations for interconnection of systems.

The assumption that a connection is a fundamental prerequisite for communication in an OSI environment permeates the Reference Model and is one of the most useful and important unifying concepts of the OSI architecture. Further study is under way to determine whether it is appropriate to include provisions for a connectionless-mode operation to expand the scope of application of the OSI Reference Model of Recommendation X.200.

This appendix has been produced to introduce the terms and concepts of the connectionless-mode of operation to facilitate the study of this provision. The two alternatives (connection-mode transmission and connectionless-mode transmission) can be treated as complementary concepts and can be applied appropriately in the different circumstances for which each is suited.

II.2 *What is connectionless-mode transmission*

Connectionless-mode transmission in one form or another has been a consideration in the specification of services and protocols for data communication. The terms "message mode", "datagram", "transaction mode" and "connection free" have been used in the literature to describe variations on the same basic theme; the transmission of a unit of data in a single self-contained operation without establishing, maintaining and releasing a connection.

Since connectionless-mode transmission and connection-mode transmission are complementary concepts, they are best understood in juxtaposition, particularly since connectionless-mode transmission is defined most easily with respect to its relationship to the concept of a connection.

II.2.1 *Connection-mode transmission*

In the formal terminology of the Reference Model, a connection is an association established for the transfer of data between two or more peer-entities. This association is established between the peer-entities themselves and between each entity and the next lower layer. The ability to establish a connection and to transfer data over it is provided to the entities in a given layer by the next lower layer as a connection-mode service. An instance of the use of a connection-mode service by peer-entities proceeds through three distinct phases:

a) connection establishment,

b) data transfer,

c) connection release.

In addition to the clearly distinguishable lifetime exhibited by these phases, a connection has the following fundamental characteristics:

i) it involves establishing and maintaining a three or more party agreement concerning the transfer of data between the peer-entities concerned and the layer providing the service;

ii) it allows the negotiation among all parties concerned of the parameters and options that will govern the transfer of data;

iii) it provides connection identification by means of which the overheads associated with address resolution and address transmission can be avoided during the data transfer phase;

iv) it provides a context within which successive units of data transferred between the peer-entities are logically related, and therefore makes it possible to maintain sequence and provide flow control.

The characteristics of connection-mode transmission are attractive in applications that call for relatively long-lived, stream-oriented interactions between entities in stable configurations. In these cases, the entities involved:

1) initially discuss their requirements and agree to the terms of their interaction, reserving whatever resources they may need;

2) transfer a series of related units of data to accomplish their mutual objective; and

3) explicitly end their interaction, releasing the previously reserved resources.

The properties of connection-mode transmission are also relevant in a wide range of other applications.

II.2.2 Connectionless-mode transmission

In formal terminology, connectionless-mode transmission is the transmission of a single unit of data from a source service-access-point to one or more destination service-access-points without establishing a connection. A connectionless-mode transmission service allows an entity to initiate such a transmission by the performance of a single service access.

In contrast to a connection, an instance of the use of the connectionless-mode service does not have a clearly distinguishable lifetime. In addition, the connectionless service, unless otherwise explicitly determined, has the following fundamental characteristics:

a) it requires only a pre-existing association between the peer-entities involved, which determines the characteristics of the data to be transmitted, and a two-party agreement between each peer-entity and the service provider; no dynamic peer-to-peer agreement is involved in an instance of the use of the service;

b) all of the information required to deliver a unit of data (destination address, quality of service selection, service options, etc.), is presented to the layer providing the connectionless-mode service, together with the user data to be transmitted, in a single access that is not required to relate to any other service access;

c) each unit of data transmitted is entirely self-contained and can be routed independently by the service provider.

Connectionless-mode transmission creates no relationship, expressed or implied, between data units. Nothing about the service invocation, the transmission of the data by the connectionless-mode service, or the data unit itself, affects or is affected by any other past, present or future operation, whether connection-mode or connectionless-mode. A series of data units handed one after the other to a connectionless-mode service for delivery to the same destination will not necessarily be delivered to the destination in the same order. The connectionless-mode service will not necessarily report or recover instances of non-delivery.

In practice, management facilities may be utilized in order to convey characteristics of the nature, quality and type of service provided by one layer to the next higher layer of the architecture prior to the invocation of that service. This facility (or set of facilities) may provide information concerning the aforementioned service characteristics in an *a priori* manner, thereby providing information which may be relied upon for subsequent invocations of the service. In contrast, the information may be provided in a dynamic fashion, invoked (perhaps) prior to each instance of use of the connectionless-mode service. Where added value may be determined prior to the use of a connectionless-mode service, parameters relating such characteristics will be related in a complementary fashion to the request for provision of the service.

II.3 *General architectural principles*

II.3.1 *Basic concepts*

In order for (N + 1)-entities to communicate using either an (N)-connection-mode service or an (N)-connectionless-mode service, there must be a prearranged association between them constituted by the *a priori* knowledge that each must have of the other(s) in order at least to initiate the use of the service. The association comprises four elements:

a) knowledge of the addresses of the peer-entities involved;

b) knowledge of a protocol agreed by the peer-entities for use at least for initial communication;

c) knowledge of the probable availability for communication of the peer-entities;

d) knowledge of the Quality of Service on offer from the (N)-service.

II.3.2 *Communication between peer-entities*

(N + 1)-entities can communicate using an (N)-connectionless-mode service provided that there is a prearranged association between them providing knowledge about each other that allows them to do so. This knowledge must allow the location of (N + 1)-entities to be determined, and it must determine the correct interpretation of (N) service data units by a receiving (N + 1)-entity. It may define the rates of transfer, rates of response, and the protocol in use between the (N + 1)-entities. The knowledge may result from a prior agreement between the (N + 1)-entities on the parameters, formats and options to be used.

There are instances where the connectionless-mode service provided by the (N + 1)-layer does not provide direct access between (N)-service-access-points supported by the layer. Connectionless-mode transmission can still occur between these service-access-points if one or more (N)-entities provide a relay. The fact that an (N + 1)-connectionless-mode transmission is relayed by one or more (N)-entities is known neither by the (N - 1) layer nor the (N + 1)-layer.

APPENDIX III
(to Recommendation X.212)

## Use of the X.25 LAPB-compatible DTE data link procedures to provide the connection-mode Data Link Service for Open Systems Interconnection for CCITT applications

This appendix defines a method for providing the OSI Connection-mode Data Link Service (CODLS) through the use of the X.25 LAPB-compatible DTE data link procedures as described in X.25 and X.75 (abbreviated to X.25/LAPB, for the remainder of this document).

The relationship between the X.25 LAPB-compatible DTE data link procedures and the OSI CODLS is shown in Figure III-1/X.212. This relationship is described only in terms of the Data Link Layer entities that provide the CDLS.

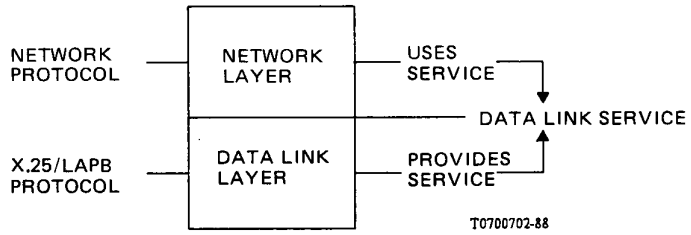


FIGURE III-1/X.212

**Relationship of the X.25 LAPB to the OSI Connection-mode Data Link Service**

III.1 *Scope and field of application*

The OSI CODLS, as stated above, is defined in terms of a set of primitive actions and events and associated parameters. For a protocol to support this service, there must be a mapping between the abstract primitives and parameters of the CODLS and the real elements of the protocol. This appendix provides such a mapping for the X.25/LAPB single link procedure (the extension of this document to multi-link procedures and other layer 2 protocols is for further study).

This appendix specifies a set of requirements applicable both to end systems and to the operation of half of an intermediate system (i.e. the Data Link Layer of an Interworking Unit/Relay Open System which operates relaying at the network layer).

III.2 *References*

In addition to the references identified in § 2 of this Recommendation, the following references are applicable to this appendix.

| | |
|---|---|
| Recommendation X.25 | 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. |
| Recommendation X.75 | Packet-switched signalling system between public networks providing data transmission services. |
| ISO 7776 | Information processing systems - Data communications - High level data link control procedure - Description of the X.25 LAPB-compatible DTE data link procedures. |

III.3 *Definitions*

III.3.1 *Reference model definitions*

The following terms, as defined in Recommendation X.200, are used in this appendix.

- Data Link Connection;
- Data Link Layer;
- Data Link Service;
- Data Link Service Access Point;
- Data Link Service Access Point address.

III.3.2 *Service convention definitions*

The following terms, as defined in Recommendation X.210, are used in this appendix:

- Data Link Service user;
- Data Link Service provider;
- primitive;
- request;
- indication;
- response;
- confirm.

III.3.3 *X.25 definitions*

The following terms, as defined in Recommendation X.25, are used in this appendix:

- Data Link;
- Data Terminal Equipment;
- Data Circuit-Terminating Equipment.

III.4 *Abbreviations*

III.4.1 *Data Link Service abbreviations*

CODLS Connection-mode Data Link Service

DLL Data Link Layer

DLSAP Data Link Service Access Point

OSI Open Systems Interconnection

QOS Quality of Service

DLC Data Link Connection

DLS Data Link Service

III.4.2 *X.25/LAPB abbreviations*

| | |
|---|---|
| LAPB | Link Access Procedure Balanced mode |
| I | Information (frame) |
| DM | Disconnected Mode (frame) |
| SABM | Set Asynchronous Balanced Mode (frame) |
| UA | Unnumbered Acknowledgement (frame) |
| FRMR | Frame Reject (frame) |
| RR | Receive Ready (frame) |
| RNR | Receive Not Ready (frame) |
| REJ | Reject (frame) |
| DTE | Data Terminal Equipment |
| DCE | Data Circuit-terminating Equipment |
| DXE | either a DTE or a DCE |
| SABME | Set Asynchronous Balanced Mode Extended |

III.5 *Overview*

The Data Link Service provides for the transparent transfer of the data between the DLS users.

III.5.1 *Elements of the X.25/LAPB used to support the OSI CODLS*

The X.25/LAPB, as defined in X.25 and X.75, provides a specific realization for the transparent transfer of the data between DLS users of the CODLS. The elements of this protocol to be considered are the frames and the fields to be mapped to the primitives and parameters of the OSI CODLS.

Table III-1/X.212 below lists the X.25/LAPB frames and associated fields used when supporting the OSI CODLS.

TABLE III-1/X.212

**Frames and fields of the X.25/LAPB used to support the OSI CODLS**

| Frame types | Fields |
|---|---|
| SABM/SABME | Address field |
| DISC<br>DM | Address field |
| I | Information field, N(R), N(S),<br>Address field |
| RR<br>RNR<br>REJ | N(R) |
| UA | Address field |
| FRMR | Information field |

*Note 1* - All user data fields are octet aligned.

*Note 2* - The address field of every type of frame is used to address the appropriate DLSAP.

*Note 3* - RR, RNR, REJ and FRMR frames are not directly mapped to the OSI CODLS primitives but are required for the correct operation of the protocol.

III.5.2 *General operation of the X.251LA PB for supporting the OSI CODLS*

The X.25/LAPB can be used to provide the OSI CODLS in an end system connected to a public or private X.25 PSDN. It can also be used in an environment where the end systems are connected via a dedicated path or via a circuit switched connection.

As shown in Figure III-2/X.212, this DLS provider (more particularly, the DLL entity in the end system) must provide a translation between:

a) the primitives and parameters of the OSI CODLS; and

b) the frames and associated fields of the X.25/LAPB.

III.6 *Data link connection establishment phase*

III.6.1 *Primitive/parameter and frame/field relationship*

Table III-2/X.212 shows the relationships between the primitives/parameters used during the Data Link Connection establishment Phase and the frames/fields associated with the Data Link Set-Up Procedure.

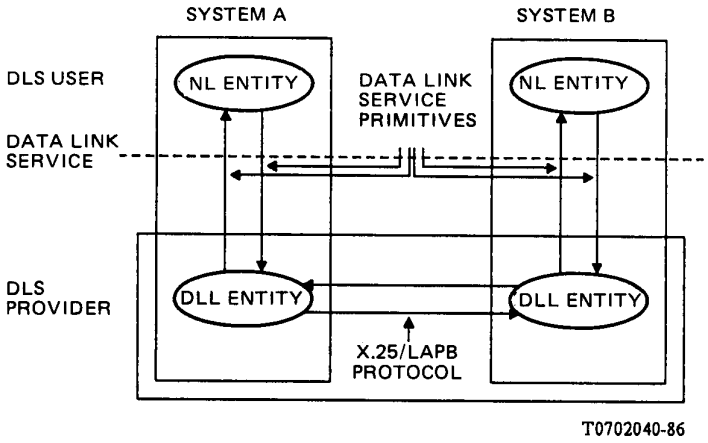


FIGURE III-2/X.212

**Relationships between the Data Link Service primitives and LAPB Protocol**

TABLE III-2/X.212

**CODLS: X.25/LAPB mapping for the Data Link Connection Establishment Phase**

| CODLS | X.25/LAPB |
|---|---|
| PRIMITIVES: | FRAMES: |
| DL-CONNECT request | SABM/SABME (see Note 2) |
| DL-CONNECT indication | SABM/SABME (see Note 2) |
| DL-CONNECT response | UA |
| DL-CONNECT confirm | UA |
| PARAMETERS: | FIELDS: |
| Called address | Address field |
| Calling address | Address field |
| Responding address | Address field |
| QOS-parameter set | None (see Note 1) |

*Note 1* - Since only one level of QOS is available, QOS-parameter set selection is not available when using X.25/LAPB to provide the OSI CODLS.

*Note 2* - The frame used is dependent of the type(s) of procedure(s) supported by the DLS provider. The relationships between the throughput QOS-parameter and the use of SABM or SABME when both are supported by the provider is for further study.

III.6.2 *Procedures*

III.6.2.1 *Primitives/frames mapping*

When a DLL entity receives a DL-CONNECT request or DL-CONNECT response primitive from a DLS user, it transmits an SABM/SABME or UA frame, respectively, across the DTE/DXE interface if it can do so before receiving a DL-DISCONNECT request. In the case of a DL-CONNECT response the procedures of 111.7.2.1.1 apply.

When a DLL entity in the disconnected phase receives an SABM/SABME frame, it signals a DLCONNECT indication to the DLS user if it can do so before receiving a DISC frame. When a DLL entity receives in the disconnected phase a UA frame in response to an SABM/SABME frame, it signals a DLCONNECT confirm to the DLS user unless it has already received a DL-DISCONNECT request from the DLS user.

III.6.2.2 *DLSAP addresses*

There is a maximum of one DLC within a DLSAP at any time. There is a one-to-one relationship between the DLC and the Physical Connection End Point. DLSAP addresses (Called, Calling and Responding) may only take one of the two values A and B as specified in X.25 and X.75.

The Called Address of a DL-CONNECT request primitive is mapped to the Address Field of the corresponding SABM/SABME frame. The Calling Address is not conveyed within the protocol.

The Address Field of a received SABM/SABME frame is mapped to the Called Address of a DLCONNECT indication primitive. The Calling Address, not conveyed within the protocol, is implicitly deduced or may be implied.

The Responding Address of a DL-CONNECT response primitive is mapped to the Address Field of the corresponding UA frame. The Responding Address is identical to the Called Address in such a point-to-point operation. The Calling Address is not conveyed within the protocol.

The Address Field of a received UA frame is mapped to the Responding Address of a DL-CONNECT confirm primitive. The Calling Address, not conveyed within the protocol, is implicitly deduced or may be implied.

III.6.2.3 *QOS parameter set*

Each X.25/LAPB DLL provides only one value for each QOS subparameter. The DLS user is supposed to have an *a priori* knowledge of the QOS supported by each underlying DLL entity (the QOS may be related to the environment: dedicated path, PSPDN, or circuit switched connection). Consequently, the choice of a DLL entity by a DLS user may be done on the basis of this *a priori* knowledge of the supported QOS.

III.7 *Data link connection release phase*

III.7.1 *Primitive/parameter and frame/field relationships*

Table III-3/X.212 shows the relationships between the primitives/parameters used during the Data Link Connection Release Phase and the frames/fields associated with the Data Link Clearing Procedures.

III.7.2 *Procedures*

III.7.2.1 *Primitive/frame mapping*

III.7.2.1.1 *Connection rejection*

When a DLL entity receives a DL-DISCONNECT request primitive as an answer to the issuance of a DL-CONNECT indication primitive or before transmitting the UA frame as a result of receiving a DLCONNECT response, it transmits a DM frame across the DTE/DXE interface.

When a DLL entity receives a DM frame in response to the transmission of an SABM/SABME frame, it issues a DL-DISCONNECT indication primitive unless it has already received a DL-DISCONNECT request from the DLS user.

TABLE III-3/X.212

**CODLS: X.25/LAPB mapping for the Data Link Connection Release Phase**

| CODLS | X.25/LAPB |
|---|---|
| PRIMITIVES:<br>DL-DISCONNECT request<br>DL-DISCONNECT indication | FRAMES:<br>DISC/DM<br>DISC/DM (see Note 2) |
| PARAMETERS:<br>Originator and reason | FIELDS:<br>one (see Note 1) |

*Note 1* - Originator is always local when using X.25/LAPB to support the OSI CODLS. Consequently, there is no need to convey a specific parameter within the protocol during the release phase.

*Note 2* - DM frame is mapped from a DL-DISCONNECT request primitive if in the connection establishment phase.

III.7.2.1.2 *Connection release from the data transfer phase*

When a DLL entity receives a DL-DISCONNECT request primitive from a DLS user, after it has transmitted a UA frame in response to an SABM/SABME frame, it transmits a DISC frame across the DTE/DXE interface, except if it has previously transmitted a DISC frame because of a protocol error (see below).

If a DL entity detects an error in the operation of the X.25/LAPB for which its action is to clear the link, it transmits a DISC frame across the DTE/DXE interface. If the link is associated with Data Link Connection, it also signals a DL-DISCONNECT indication primitive to the DLS user.

When a DLL entity receives a DISC frame it signals a DL-DISCONNECT indication to the DLS user.

*Note* - When a DLL entity receives, from the DLS user, a DL-CONNECT request primitive following a DL-DISCONNECT request primitive, the issuing of the SABM/SABME frame is postponed until the DISC frame, related to this DL-DISCONNECT request primitive, has been acknowledged or repeated N2 times.

III.7.2.2 *Originator/Reason*

Originator and Reason parameters have only local significance when using a X.25/LAPB to support the OSI CODLS. They are not conveyed by specific parameters within the protocol during the release phase.

Upon the reception of a DISC or a DM frame:

a) the value of the Originator parameter in the DL-DISCONNECT indication primitive is "unknown";

and

b) the value of the Reason parameter in the DL-DISCONNECT indication primitive is always "reason unspecified".

If the DISC or DM frame is locally issued by the provider:

i) the value of the Originator parameter in the DL-DISCONNECT indication primitive is "provider"; and

ii) the value of the Reason parameter in the DL-DISCONNECT indication primitive is one of those contained in § 13.2.2.a) of this Recommendation.

Consequently, the meaning of the Originator and Reason parameters in a DL-DISCONNECT request primitive is not guaranteed to be the same as in the related DL-DISCONNECT indication primitive.

III.8 *Data transfer phase - Data transfer service*

III.8.1 *Primitive/parameter and frame/field relationships*

Table III-4/X.212 shows the relationships between the primitives/parameters used for the Data Transfer Service and the frames/fields associated with the Data Transfer Procedures.

TABLE III-4/X.212

**CODLS: X.25/LAPB mapping for the Data Transfer Service**

| CODLS | X.25/LAPB |
|---|---|
| PRIMITIVES:<br>DL-DATA request<br>DL-DATA indication | FRAMES:<br>1<br>1 |
| PARAMETERS:<br>DLS user data | FIELDS:<br>Information field |

III.8.2 *Procedures*

III.8.2.1 *Primitives/frames mapping*

When a DLL entity receives a DL-DATA request primitive from a DLS user, it transmits a new I frame across the DTE/DXE interface.

When a DLL entity receives an I frame with N(S) - V(R), it signals a DL-DATA indication primitive to the DLS user.

III.8.2.2 *DLS User Data*

The Data Field of the I frame is directly mapped to the User Data parameters of the DL-DATA request and DL-DATA indication primitives.

III.9 *Data transfer phase - reset service*

III.9.1 *Primitive/parameter and frame/field relationships*

Table III-5/X.212 shows the relationships between the primitives/parameters used for the reset service and the frames/fields associated with the Reset Procedures.

III.9.2 *Procedures*

III.9.2.1 *Primitive/frames mapping*

When a DLL entity receives a DL-RESET request primitive from a DLS user, it transmits an SABM/SABME frame across the DTE/DXE interface.

When a DLL entity receives in the user initiated reset phase a UA frame in response to an SABM/SABME frame, it signals a DL-RESET confirm to the DLS user if it has not already done so (see note to Table III-5/X.212).

TABLE III-5/X.212

**CODLS: X.25/LAPB mapping for the Data Link Reset Service**

| CODLS | X.25/LAPB |
|---|---|
| PRIMITIVES:<br>DL-RESET request<br>DL-RESET indication<br>DL-RESET response<br>DL-RESET confirm | FRAMES:<br>SABM/SABME<br>SABM/SABME<br>UA<br>UA (see Note 1) |
| PARAMETERS:<br>Originator and reason | FIELDS: |

*Note 1* - Since there is no requirement for a fixed time relationship between the response and confirm primitives, the DLS provider can issue the confirm primitive before receiving the UA frame.

If a DLL entity detects an error in the operation of the X.25/LAPB for which its action is to re-establish the link, it transmits an SABM/SABME frame across the DTE/DXE interface. If the link is associated with a data link connection, it also signals a DL-RESET indication primitive to the DLS user.

When a DLL entity receives, in the local provider initiated reset phase, a UA frame in response to an SABM/SABME frame, no primitive is issued to the DLS user.

When a DLL entity receives an SABM/SABME frame during the data transfer phase, it issues a DL-RESET indication primitive.

When a DLL entity receives a DL-RESET response from a DLS user it transmits a UA frame across the DTE/DXE interface, unless the reset was locally generated.

III.9.2.2 *Originator/Reason*

Originator and Reason parameters have only local significance when using X.25/LAPB to support the OSI CODLS. They are not conveyed by specific parameters within the protocol.

Upon the reception of a DISC frame:

a) the value of the Originator parameter in the DL-RESET indication primitive is "unknown";

b) the value of the Reason parameter in the DL-RESET indication primitive is always "reason unspecified".

If the DISC frame is locally issued by the provider:

i) the value of the Originator parameter in the DL-RESET indication primitive is "provider"; and

ii) the value of the Reason parameter in the DL-RESET indication primitive is data link flow control congestion, or data link error as appropriate.

Consequently, the meaning of Originator and Reason parameters in a DL-RESET request primitive is not guaranteed to be the same as in the related DL-RESET indication primitive.