



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

UIT-T

SECTOR DE NORMALIZACIÓN
DE LAS TELECOMUNICACIONES
DE LA UIT

Q.832.3

(01/2001)

SERIE Q: CONMUTACIÓN Y SEÑALIZACIÓN
Interfaz Q3

Coordinación del acceso en banda ancha

Recomendación UIT-T Q.832.3

(Anteriormente Recomendación del CCITT)

RECOMENDACIONES UIT-T DE LA SERIE Q
CONMUTACIÓN Y SEÑALIZACIÓN

SEÑALIZACIÓN EN EL SERVICIO MANUAL INTERNACIONAL	Q.1–Q.3
EXPLOTACIÓN INTERNACIONAL SEMIAUTOMÁTICA Y AUTOMÁTICA	Q.4–Q.59
FUNCIONES Y FLUJOS DE INFORMACIÓN PARA SERVICIOS DE LA RDSI	Q.60–Q.99
CLÁUSULAS APLICABLES A TODOS LOS SISTEMAS NORMALIZADOS DEL UIT-T	Q.100–Q.119
ESPECIFICACIONES DE LOS SISTEMAS DE SEÑALIZACIÓN N.º 4 Y N.º 5	Q.120–Q.249
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 6	Q.250–Q.309
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R1	Q.310–Q.399
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN R2	Q.400–Q.499
CENTRALES DIGITALES	Q.500–Q.599
INTERFUNCIONAMIENTO DE LOS SISTEMAS DE SEÑALIZACIÓN	Q.600–Q.699
ESPECIFICACIONES DEL SISTEMA DE SEÑALIZACIÓN N.º 7	Q.700–Q.799
INTERFAZ Q3	Q.800–Q.849
SISTEMA DE SEÑALIZACIÓN DIGITAL DE ABONADO N.º 1	Q.850–Q.999
RED MÓVIL TERRESTRE PÚBLICA	Q.1000–Q.1099
INTERFUNCIONAMIENTO CON SISTEMAS MÓVILES POR SATÉLITE	Q.1100–Q.1199
RED INTELIGENTE	Q.1200–Q.1699
REQUISITOS Y PROTOCOLOS DE SEÑALIZACIÓN PARA IMT-2000	Q.1700–Q.1799
RED DIGITAL DE SERVICIOS INTEGRADOS DE BANDA ANCHA (RDSI-BA)	Q.2000–Q.2999

Para más información, véase la Lista de Recomendaciones del UIT-T.

Coordinación del acceso en banda ancha

Resumen

Esta Recomendación especifica la interfaz X entre el sistema de operaciones de un nodo de servicio y el sistema de operaciones de una red de acceso para la coordinación de la gestión asociada con las interfaces de tráfico VB5.1 y VB5.2 y las interfaces VB5 Q3.

Orígenes

La Recomendación UIT-T Q.832.3, preparada por la Comisión de Estudio 4 (2001-2004) del UIT-T, fue aprobada por el procedimiento de la Resolución 1 de la AMNT el 19 de enero de 2001.

PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. Este órgano estudia los aspectos técnicos, de explotación y tarifarios y publica Recomendaciones sobre los mismos, con miras a la normalización de las telecomunicaciones en el plano mundial.

La Asamblea Mundial de Normalización de las Telecomunicaciones (AMNT), que se celebra cada cuatro años, establece los temas que han de estudiar las Comisiones de Estudio del UIT-T, que a su vez producen Recomendaciones sobre dichos temas.

La aprobación de Recomendaciones por los Miembros del UIT-T es el objeto del procedimiento establecido en la Resolución 1 de la AMNT.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI.

NOTA

En esta Recomendación, la expresión "Administración" se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

PROPIEDAD INTELECTUAL

La UIT señala a la atención la posibilidad de que la utilización o aplicación de la presente Recomendación suponga el empleo de un derecho de propiedad intelectual reivindicado. La UIT no adopta ninguna posición en cuanto a la demostración, validez o aplicabilidad de los derechos de propiedad intelectual reivindicados, ya sea por los miembros de la UIT o por terceros ajenos al proceso de elaboración de Recomendaciones.

En la fecha de aprobación de la presente Recomendación, la UIT no ha recibido notificación de propiedad intelectual, protegida por patente, que puede ser necesaria para aplicar esta Recomendación. Sin embargo, debe señalarse a los usuarios que puede que esta información no se encuentre totalmente actualizada al respecto, por lo que se les insta encarecidamente a consultar la base de datos sobre patentes de la TSB.

© UIT 2001

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

ÍNDICE

Página

1	Introducción	1
1.1	Finalidad y alcance	1
2	Referencias.....	1
3	Definiciones, abreviaturas y convenios	1
3.1	Definiciones	1
3.1.1	Recursos VB5.....	1
3.2	Abreviaturas	2
3.3	Convenios	2
4	Visión general	3
4.1	Modelos de relaciones entre entidades	4
4.1.1	Diagrama de relaciones entre entidades.....	4
4.2	Jerarquía de herencia	5
5	Definiciones formales.....	5
5.1	Clases de objetos	5
5.1.1	Notas sobre perfiles para clases importadas.....	5
5.1.2	Definición de clases	5
5.2	Vinculaciones de nombre.....	7
5.2.1	xvb5-managedElementR1	7
5.3	Definición de lotes.....	7
5.4	Definición de atributos	7
5.4.1	xvb5Id (X-VB5 identifier)	7
5.5	Definición de acciones.....	7
5.5.1	addAnLoopRequest (add AN loop request).....	7
5.5.2	addLupsRequest (add LUPs request).....	8
5.5.3	addVb5ConnectionRequest (add VB5 connection request).....	8
5.5.4	addVb5InterfaceRequest (add VB5 interface request)	8
5.5.5	addVb5ProtocolRequest (add VB5 protocol request)	8
5.5.6	addVb5ProtocolVpRequest (add VB5 protocol VP request).....	9
5.5.7	addVb5VcsRequest (add VB5 VCs request)	9
5.5.8	addVb5VpsRequest (add VB5 VPs request).....	9
5.5.9	anServiceLabelInquiry (AN service label inquiry)	9
5.5.10	auditVb5ConnectionRequest (audit VB5 connection request)	9
5.5.11	auditVb5VpciRequest (audit VB5 VPCI request).....	10
5.5.12	listLupsRequest (list LUPs request)	10
5.5.13	listVb5ProtocolDetailsRequest (list protocol details request)	10
5.5.14	listVb5InterfacesRequest (list VB5 interfaces request).....	10

5.5.15	listVb5VcsRequest (list VB5 VCs request)	11
5.5.16	listVb5VpsRequest (list VB5 VPs request)	11
5.5.17	removeAnLoopRequest (remove AN loop request).....	11
5.5.18	removeLupsRequest (remove LUPs request).....	11
5.5.19	removeVb5ConnectionRequest (remove VB5 connection request).....	11
5.5.20	removeVb5InterfaceRequest (remove VB5 interface request).....	12
5.5.21	removeVb5ProtocolRequest (remove VB5 protocol request)	12
5.5.22	removeVb5ProtocolVpRequest (remove VB5 protocol Vp request).....	12
5.5.23	removeVb5VcsRequest (remove VB5 VCs request)	12
5.5.24	removeVb5VpsRequest (remove VB5 VPs request).....	13
5.5.25	snAccessLabelsInquiry (SN access labels inquiry)	13
5.6	Definición de notificaciones	13
5.6.1	addLupsIndication (add LUPs indication)	13
5.6.2	addVb5ConnectionIndication (add VB5Connection indication)	13
5.6.3	addVb5InterfaceIndication (add VB5 interface indication).....	13
5.6.4	addVb5ProtocolIndication (add VB5 protocol indication).....	14
5.6.5	addVb5ProtocolVpIndication (add VB5 protocol VP indication)	14
5.6.6	addVb5VcsIndication (add VB5 VCs indication).....	14
5.6.7	addVb5VpsIndication (add VB5 VPs indication)	14
5.6.8	removeLupsIndication (remove LUPs indication)	14
5.6.9	removeVb5ConnectionIndication (remove VB5Connection indication)	15
5.6.10	removeVb5InterfaceIndication (remove VB5 interface indication).....	15
5.6.11	removeVb5ProtocolIndication (remove VB5 protocol indication).....	15
5.6.12	removeVb5ProtocolVpIndication (remove VB5 protocol VP indication) ...	15
5.6.13	removeVb5VcsIndication (remove VB5 VCs indication)	15
5.6.14	removeVb5VpsIndication (remove VB5 VPs indication)	16
5.6.15	resourceStatusIndication (resource status indication)	16
6	Definiciones de tipos	16
7	Pilas de protocolos.....	23
Anexo A	– Necesidades de gestión.....	24
A.1	Necesidades de gestión de carácter general.....	24
A.1.1	Coordinación de operaciones relativas a la configuración de los VP y VC .	24
A.1.2	Comprobación de la VPC	24
A.1.3	Coordinación de datos de configuración de puerto	24
A.1.4	Coordinación de valores de VPCI	24
A.1.5	Coherencia de la configuración	24
A.1.6	Disponibilidad de la información	24

	Página
A.2 Coordinador de la interfaz VB5	24
A.2.1 Creación	24
A.2.2 Verificación y auditoría	25
A.2.3 Modificación	25
A.2.4 Supresión.....	25
A.2.5 Provisión de VC y VP.....	25
A.2.6 Verificación y auditoría	26
A.2.7 Modificación de VP y VC.....	26
A.2.8 Verificación y auditoría	26
A.2.9 Supresión de VP y VC	26
A.3 Coordinación de la interfaz UNI	26
A.3.1 Creación	26
A.3.2 Verificación y auditoría	26
A.3.3 Modificación	26
A.3.4 Supresión.....	26
A.4 Necesidades de coordinación del control de conexión portadora de banda ancha	27
A.4.1 VC en la interfaz VB5	27
A.4.2 VC en la interfaz UNI.....	27
A.5 Gestión de las averías y de la calidad de funcionamiento	28
A.5.1 Informes de averías.....	28
A.5.2 Localización de averías.....	28
Anexo B – Necesidades de transacciones	28
Apéndice I – Bibliografía	48
Apéndice II – Sumario de transacciones	50

Coordinación del acceso en banda ancha

1 Introducción

1.1 Finalidad y alcance

Esta Recomendación especifica la interfaz X entre el sistema de operaciones (OS, *operations system*) de un nodo de servicio (SN, *service node*) y el sistema de operaciones de una red de acceso (AN, *access network*) para la coordinación de la gestión asociada con las interfaces de tráfico VB5.1 y VB5.2 [2] y [3] y las interfaces VB5 Q3 [4] y [5].

En la medida de lo posible, se utilizan protocolos existentes y el trabajo se centra en la definición del modelo de objeto. La definición de la funcionalidad del sistema de operaciones de la RGT está fuera del ámbito de esta Recomendación.

La gestión de la seguridad también está fuera del ámbito de esta Recomendación.

2 Referencias

Las siguientes Recomendaciones del UIT-T y otras referencias contienen disposiciones que, mediante su referencia en este texto, constituyen disposiciones de la presente Recomendación. Al efectuar esta publicación, estaban en vigor las ediciones indicadas. Todas las Recomendaciones y otras referencias son objeto de revisiones por lo que se preconiza que los usuarios de esta Recomendación investiguen la posibilidad de aplicar las ediciones más recientes de las Recomendaciones y otras referencias citadas a continuación. Se publica periódicamente una lista de las Recomendaciones UIT-T actualmente vigentes.

- [1] UIT-T G.902 (1995), *Recomendación marco sobre redes de acceso funcional – Arquitectura y funciones, tipos de accesos, gestión y aspectos del nodo de servicio.*
- [2] UIT-T G.967.1 (1998), *Interfaces V en el nodo de servicio – Especificación del punto de referencia VB5.1.*
- [3] UIT-T G.967.2 (1999), *Interfaces V en el nodo de servicio – Especificación del punto de referencia VB5.2.*
- [4] UIT-T Q.832.1 (1998), *Gestión de la interfaz VB5.1.*
- [5] UIT-T Q.832.2 (1998), *Gestión de la interfaz VB5.2.*
- [6] UIT-T X.721 (1992) | ISO/CEI 10165-2:1992, *Tecnología de la información – Interconexión de sistemas abiertos – Estructura de la información de gestión: Definición de la información de gestión.*
- [7] IETF RFC 1057(1988): *RPC: Remote Procedure Call Protocol Specification: Version 2.*

3 Definiciones, abreviaturas y convenios

3.1 Definiciones

En esta Recomendación se definen los términos siguientes.

3.1.1 Recursos VB5: La gestión de funciones de puerto de usuario y de funciones de puerto de servicio que proporcionan las funcionalidades de interfaz usuario-red (UNI, *user network interface*) e interfaz de nodo de servicio (SNI, *service node interface*), respectivamente, se consideran en esta Recomendación dentro del marco definido en UIT-T G.902 [1]. Los recursos específicos de la

transmisión están fuera de su alcance. En la presente Recomendación, los recursos VB5 se designan brevemente por recursos.

En la presente Recomendación se utilizan asimismo los siguientes términos definidos en otras Recomendaciones UIT-T:

UIT-T G.902 [1]: Red de acceso (AN), funciones de puerto de usuario, nodo de servicio (SN), interfaz de nodo de servicio (SNI), funciones de puerto de servicio.

UIT-T G.967.1 [2]: Puerto de servicio lógico (LSP, *logical service port*), puerto de usuario lógico (LUP, *logical user port*), puerto de servicio físico (PSP), puerto de usuario físico (PUP), coordinación de gestión en tiempo real (RTMC, *real time management coordination*), puerto de usuario virtual (VUP).

UIT-T G.967.2 [3]: Control de conexión portadora de banda ancha (B-BCC).

3.2 Abreviaturas

En esta Recomendación se utilizan las siguientes siglas.

AN Red de acceso (*access network*)

ASN.1 Notación de sintaxis abstracta uno (*abstract syntax notation one*)

ATM Modo de transferencia asíncrono (*asynchronous transfer mode*)

B-BCC Control de conexión portadora de banda ancha (*broadband bearer connection control*)

IETF Grupo especial de ingeniería en Internet (*Internet engineering task force*)

LSP Puerto de servicio lógico (*logical service port*)

LUP Puerto de usuario lógico (*logical user port*)

OS Sistema de operaciones (*operations system*)

RFC Petición de comentarios (*request for comment*)

RGT Red de gestión de las telecomunicaciones

RTMC Coordinación de gestión en tiempo real (*real time management coordination*)

SN Nodo de servicio (*service node*)

SNI Interfaz de nodo de servicio (*service node interface*)

TTP Punto de terminación de camino (*trail termination point*)

UNI Interfaz usuario-red (*user-network interface*)

VC Canal virtual (*virtual channel*)

VP Trayecto virtual (*virtual path*)

VPC Conexión de trayecto virtual (*virtual path connection*)

VPCI Identificador de conexión de trayecto virtual (*virtual path connection identifier*)

3.3 Convenios

Los nombres de los objetos y sus características y la ASN.1 asociada que aquí se definen se escriben con su letra inicial en mayúscula para indicar el comienzo de la palabra siguiente, y los acrónimos se tratan como si fueran palabras.

En toda esta Recomendación, la denominación de los nuevos atributos se efectúa de acuerdo con las siguientes directrices:

- El nombre de un atributo finaliza con la cadena "Ptr" si, y solamente si, el valor del atributo está destinado a identificar un solo objeto.
- El nombre de un atributo termina con la cadena "PtrList" si, y solamente si, el valor del atributo está destinado a identificar uno o más objetos.
- El nombre de un atributo se compone del nombre de una clase de objeto seguido de la cadena "Ptr" si, y solamente si, el valor del atributo está destinado a identificar una determinada clase de objeto.
- Si un atributo está destinado a identificar diferentes clases de objetos, se le da un nombre descriptivo y en el comportamiento del atributo se proporciona una descripción.
- El nombre de un atributo finaliza con la cadena "Id" si, y solamente si, el valor del atributo está destinado a identificar el nombre de un objeto, en cuyo caso este atributo deberá ser el primero indicado, deberá utilizar el NameType de la ASN.1, y no se deberá utilizar para llevar otra información.
- El nombre de un atributo se compone del nombre de una clase de objeto seguido de la cadena "Id" si, y solamente si, el valor del atributo está destinado a identificar el nombre de la clase de objeto que tiene ese atributo.

4 Visión general

Los siguientes diagramas del modelo de información tienen por finalidad aclarar las relaciones entre las diferentes clases de objetos del modelo.

- 1) Diagramas de modelos de relaciones entre entidades, que muestran las relaciones entre los diferentes objetos gestionados.
- 2) Diagramas de jerarquía de herencia, que muestran la forma en que los objetos gestionados se derivan unos de otros (es decir, los diferentes trayectos de las características de los diferentes objetos gestionados que han sido heredadas).

Los diagramas sólo tienen por cometido aclarar la exposición. La especificación formal mediante plantillas de las directrices para la definición de objetos gestionados (GDMO, *guidelines for the definition of managed objects*) y las definiciones de tipo ASN.1 constituyen la información que interesa a efectos de las implementaciones.

4.1 Modelos de relaciones entre entidades

En los diagramas se utilizan los siguientes convenios:

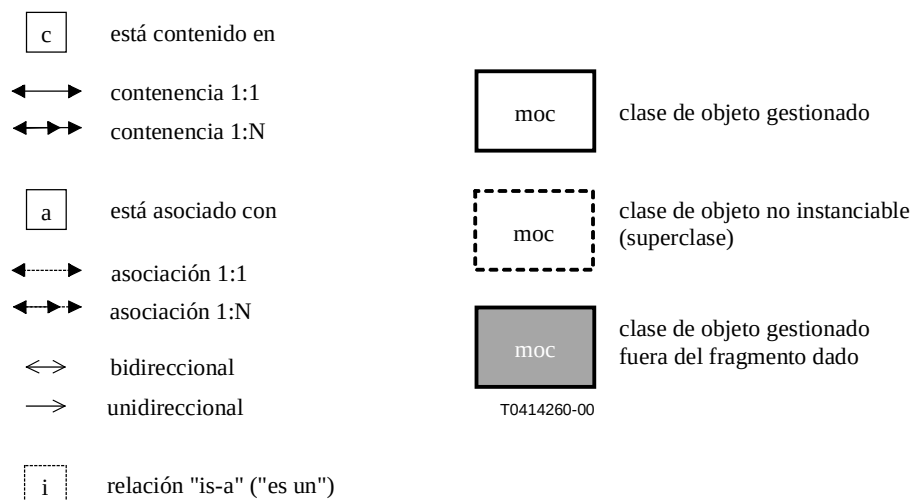


Figura 1/Q.832.3 – Convenios utilizados en los diagramas para los modelos de relaciones entre entidades

Cuando la direccionalidad de la contenencia no está clara puede ser determinada por implicaciones ya que la clase raíz es única.

4.1.1 Diagrama de relaciones entre entidades

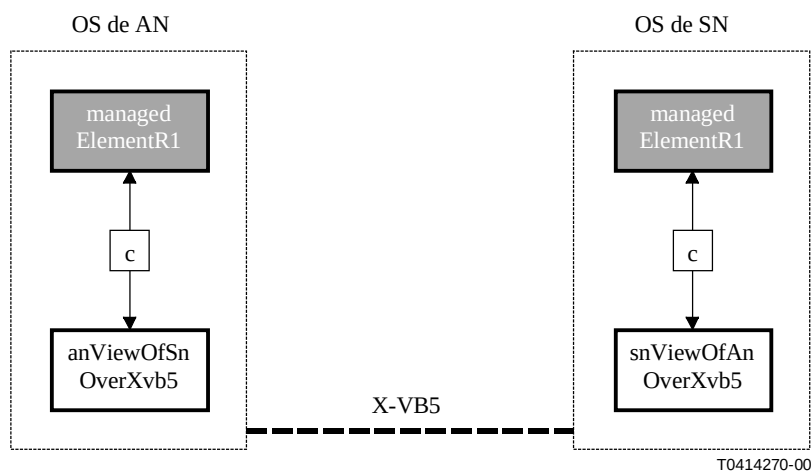


Figura 2/Q.832.3 – Diagrama de relaciones entre entidades

4.2 Jerarquía de herencia

En la figura 3 se muestran las relaciones de herencia desde el objeto de nivel más alto ("top") de UIT-T X.721 hasta los objetos gestionados que se definen en esta Recomendación.

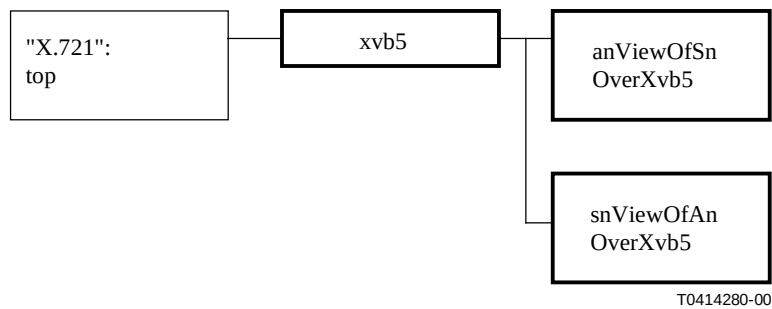


Figura 3/Q.832.3 – Jerarquía de herencia

5 Definiciones formales

En esta cláusula se presentan las definiciones formales de las clases de objetos gestionados, vinculaciones de nombre, lotes generales, comportamientos, atributos, acciones y notificaciones.

5.1 Clases de objetos

En esta cláusula se especifican las clases de objetos para todos los objetos gestionados utilizados en el modelo de información de gestión. Estas clases de objetos se definen, sea en la presente especificación, sea por referencia a otras especificaciones. Las clases de objetos gestionados que están definidas en otros lugares y que sólo se utilizan a efectos de contención no se han incluido, pero se identifican mediante las vinculaciones de nombre para las clases aquí especificadas.

5.1.1 Notas sobre perfiles para clases importadas

No hay clases importadas.

5.1.2 Definición de clases

5.1.2.1 anViewOfSnOverXvb5 (AN view of SN over X-VB5)

```
anViewOfSnOverXvb5 MANAGED OBJECT CLASS
  DERIVED FROM xvb5;
  CHARACTERIZED BY
    anViewOfSnOverXvb5Pkg PACKAGE
      BEHAVIOUR anViewOfSnOverXvb5Beh;
      ACTIONS
        anServiceLabelInquiry;;
REGISTERED AS {q832-3ManagedObjectClass 1};
```

```
anViewOfSnOverXvb5Beh BEHAVIOUR
  DEFINED AS
```

"This managed object represents the Service Node side of an X-VB5 interface, as seen by the Operations System of the Access Network.";

5.1.2.2 snViewOfAnOverXvb5 (SN view of AN over X-VB5)

snViewOfAnOverXvb5 MANAGED OBJECT CLASS

DERIVED FROM xvb5;

CHARACTERIZED BY

snViewOfAnOverXvb5Pkg PACKAGE

BEHAVIOUR snViewOfAnOverXvb5Beh;

ACTIONS

addAnLoopRequest,
snAccessLabelsInquiry,
removeAnLoopRequest;;

REGISTERED AS {q832-3ManagedObjectClass 2};

snViewOfAnOverXvb5Beh BEHAVIOUR

DEFINED AS

"This managed object represents the Access Network side of an X-VB5 interface, as seen by the Operations System of the Service Node.";

5.1.2.3 xvb5 (X-VB5)

xvb5 MANAGED OBJECT CLASS

DERIVED FROM "Rec. X.721|ISO/IEC 10165-2":top;

CHARACTERIZED BY

"ITU-T M.3100":operationalStatePackage,

xvb5Pkg PACKAGE

BEHAVIOUR xvb5Beh;

ATTRIBUTES

xvb5Id

GET SET-BY-CREATE,

" Rec. X.721 | ISO/IEC 10165-2": administrativeState

GET-REPLACE;

ACTIONS

addLupsRequest,
addVb5ConnectionRequest,
addVb5InterfaceRequest,
addVb5ProtocolRequest,
addVb5ProtocolVpRequest,
addVb5VcsRequest,
addVb5VpsRequest,
auditVb5ConnectionRequest,
auditVb5VpciRequest,
listLupsRequest,
listVb5ProtocolDetailsRequest,
listVb5InterfacesRequest,
listVb5VcsRequest,
listVb5VpsRequest,
removeLupsRequest,
removeVb5ConnectionRequest,
removeVb5InterfaceRequest,
removeVb5ProtocolRequest,
removeVb5ProtocolVpRequest,
removeVb5VcsRequest,
removeVb5VpsRequest;

NOTIFICATIONS

addLupsIndication,
addVb5ConnectionIndication,
addVb5InterfaceIndication,
addVb5ProtocolIndication,
addVb5ProtocolVpIndication,
addVb5VcsIndication,

```

        addVb5VpsIndication,
        removeLupsIndication,
        removeVb5ConnectionIndication,
        removeVb5InterfaceIndication,
        removeVb5ProtocolIndication,
        removeVb5ProtocolVpIndication,
        removeVb5VcsIndication,
        removeVb5VpsIndication,
        resourceStatusIndication,
        " Rec. X.721 | ISO/IEC 10165-2": stateChange,
        " Rec. X.721 | ISO/IEC 10165-2": objectCreation,
        " Rec. X.721 | ISO/IEC 10165-2": objectDeletion;;;
REGISTERED AS {q832-3ManagedObjectClass 3};

xvb5Beh BEHAVIOUR
  DEFINED AS
    "The xvb5 managed object class represents the aspects of an X-VB5 interface that are common to both sides.
    The xvb5 class is not instantiated.";

```

5.2 Vinculaciones de nombre

5.2.1 xvb5-managedElementR1

```

xvb5-managedElementR1 NAME BINDING
  SUBORDINATE OBJECT CLASS xvb5 AND SUBCLASSES;
  NAMED BY SUPERIOR OBJECT CLASS "ITU-T Rec. M.3100":managedElementR1 AND SUBCLASSES;
  WITH ATTRIBUTE xvb5Id;
  CREATE
    WITH-REFERENCE-OBJECT,
    WITH-AUTOMATIC-INSTANCE-NAMING;
  DELETE
    DELETES-CONTAINED-OBJECTS;
REGISTERED AS {q832-3NameBinding 1};

```

5.3 Definición de lotes

No se definen lotes.

5.4 Definición de atributos

5.4.1 xvb5Id (X-VB5 identifier)

```

xvb5Id ATTRIBUTE
  WITH ATTRIBUTE SYNTAX Q832-3ASN1Module.NameType;
  MATCHES FOR EQUALITY;
  BEHAVIOUR xvb5IdBeh;
REGISTERED AS {q832-3Attribute 1};

```

```

xvb5IdBeh BEHAVIOUR
  DEFINED AS
    "This attribute is used for naming instances of the managed object class xvb5 and subclasses.";

```

5.5 Definición de acciones

5.5.1 addAnLoopRequest (add AN loop request)

```

addAnLoopRequest ACTION
  BEHAVIOUR addAnLoopRequestBeh;
  MODE CONFIRMED;

```

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddAnLoopRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AddAnLoopRequestResult;
REGISTERED AS {q832-3Action 1};

addAnLoopRequestBeh BEHAVIOUR
DEFINED AS

"This action is used by the OS of the SN to request the OS of the AN to loop a connection so that cells sent to the AN will be returned.";

5.5.2 addLupsRequest (add LUPs request)

addLupsRequest ACTION
BEHAVIOUR addLupsRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddLupsRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AddLupsRequestResult;
REGISTERED AS {q832-3Action 2};

addLupsRequestBeh BEHAVIOUR
DEFINED AS

"This action is used to request the peer OS to add Logical User Ports to a VB5 interface.";

5.5.3 addVb5ConnectionRequest (add VB5 connection request)

addVb5ConnectionRequest ACTION
BEHAVIOUR addVb5ConnectionRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5ConnectionRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5ConnectionRequestResult;
REGISTERED AS {q832-3Action 3};

addVb5ConnectionRequestBeh BEHAVIOUR
DEFINED AS

"This action is used to request the peer OS to add a connection associated with a VB5 interface. The egress direction is out of the Access Network towards Service Node. The ingress direction is into the Access Network from the Service Node.";

5.5.4 addVb5InterfaceRequest (add VB5 interface request)

addVb5InterfaceRequest ACTION
BEHAVIOUR addVb5InterfaceRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5InterfaceRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5InterfaceRequestResult;
REGISTERED AS {q832-3Action 4};

addVb5InterfaceRequestBeh BEHAVIOUR
DEFINED AS

"This action is used to request the peer OS to add a VB5 interface.";

5.5.5 addVb5ProtocolRequest (add VB5 protocol request)

addVb5ProtocolRequest ACTION
BEHAVIOUR addVb5ProtocolRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5ProtocolRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5ProtocolRequestResult;
REGISTERED AS {q832-3Action 5};

addVb5ProtocolRequestBeh BEHAVIOUR
DEFINED AS
"This action is used to request the peer OS to add a protocol to an existing VB5 interface.";

5.5.6 addVb5ProtocolVpRequest (add VB5 protocol VP request)

addVb5ProtocolVpRequest ACTION
BEHAVIOUR addVb5ProtocolVpRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5ProtocolVpRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5ProtocolVpRequestResult;
REGISTERED AS {q832-3Action 6};

addVb5ProtocolVpRequestBeh BEHAVIOUR
DEFINED AS
"This action is used to request the peer OS to add a protocol VP to a VB5 interface.";

5.5.7 addVb5VcsRequest (add VB5 VCs request)

addVb5VcsRequest ACTION
BEHAVIOUR addVb5VcsRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5VcsRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5VcsRequestResult;
REGISTERED AS {q832-3Action 7};

addVb5VcsRequestBeh BEHAVIOUR
DEFINED AS
"This action is used to request the peer OS to add VCs to a VP which is associated with a VB5 interface.";

5.5.8 addVb5VpsRequest (add VB5 VPs request)

addVb5VpsRequest ACTION
BEHAVIOUR addVb5VpsRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5VpsRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5VpsRequestResult;
REGISTERED AS {q832-3Action 8};
addVb5VpsRequestBeh BEHAVIOUR
DEFINED AS
"This action is used to request the peer OS to add VPs that are associated with a VB5 interface.";

5.5.9 anServiceLabelInquiry (AN service label inquiry)

anServiceLabelInquiry ACTION
BEHAVIOUR anServiceLabelInquiryBeh;
MODE CONFIRMED;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AnServiceLabelInquiryResult;
REGISTERED AS {q832-3Action 9};

anServiceLabelInquiryBeh BEHAVIOUR
DEFINED AS
"This action is used by the OS of an AN to inquire the label that an SN uses for the AN.";

5.5.10 auditVb5ConnectionRequest (audit VB5 connection request)

auditVb5ConnectionRequest ACTION
BEHAVIOUR auditVb5ConnectionRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AuditVb5ConnectionRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AuditVb5ConnectionRequestResult;

REGISTERED AS {q832-3Action 10};

auditVb5ConnectionRequestBeh BEHAVIOUR
DEFINED AS

"This action is used to request the peer OS to audit a connection which is associated with a VB5 interface.";

5.5.11 auditVb5VpciRequest (audit VB5 VPCI request)

auditVb5VpciRequest ACTION

BEHAVIOUR auditVb5VpciRequestBeh;

MODE CONFIRMED;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AuditVb5VpciRequestInfo;

WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.AuditVb5VpciRequestResult;

REGISTERED AS {q832-3Action 11};

auditVb5VpciRequestBeh BEHAVIOUR

DEFINED AS

"This action is used to request the peer OS to audit a VPCI which is associated with a VB5 interface.";

5.5.12 listLupsRequest (list LUPs request)

listLupsRequest ACTION

BEHAVIOUR listLupsRequestBeh;

MODE CONFIRMED;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.ListLupsRequestInfo;

WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.ListLupsRequestResult;

REGISTERED AS {q832-3Action 12};

listLupsRequestBeh BEHAVIOUR

DEFINED AS

"This action is used to request the peer OS to list the Logical User Ports associated with a VB5 interface between an Access Network and a Service Node which the two Operations Systems together control.";

5.5.13 listVb5ProtocolDetailsRequest (list protocol details request)

listVb5ProtocolDetailsRequest ACTION

BEHAVIOUR listVb5ProtocolDetailsRequestBeh;

MODE CONFIRMED;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.ListVb5ProtocolDetailsRequestInfo;

WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.ListVb5ProtocolDetailsRequestResult;

REGISTERED AS {q832-3Action 13};

listVb5ProtocolDetailsRequestBeh BEHAVIOUR

DEFINED AS

"This action is used to request the peer OS to list the details of the protocols of a VB5 interface between an Access Network and a Service Node which the two Operations Systems together control.";

5.5.14 listVb5InterfacesRequest (list VB5 interfaces request)

listVb5InterfacesRequest ACTION

BEHAVIOUR listVb5InterfacesRequestBeh;

MODE CONFIRMED;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.ListVb5InterfacesRequestInfo;

WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.ListVb5InterfacesRequestResult;

REGISTERED AS {q832-3Action 14};

listVb5InterfacesRequestBeh BEHAVIOUR

DEFINED AS

"This action is used to request the peer OS to list the identities of the VB5 interfaces between Access Network(s) and the Service Node(s) which the two Operations Systems together control.";

5.5.15 listVb5VcsRequest (list VB5 VCs request)

listVb5VcsRequest ACTION
BEHAVIOUR listVb5VcsRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.ListVb5VcsRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.ListVb5VcsRequestResult;
REGISTERED AS {q832-3Action 15};

listVb5VcsRequestBeh BEHAVIOUR
DEFINED AS
"This action is used to request the peer OS to list the VCs associated with a VB5 interface.";

5.5.16 listVb5VpsRequest (list VB5 VPs request)

listVb5VpsRequest ACTION
BEHAVIOUR listVb5VpsRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.ListVb5VpsRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.ListVb5VpsRequestResult;
REGISTERED AS {q832-3Action 16};

listVb5VpsRequestBeh BEHAVIOUR
DEFINED AS
"This action is used to request the peer OS to list the VPs associated with a VB5 interface.";

5.5.17 removeAnLoopRequest (remove AN loop request)

removeAnLoopRequest ACTION
BEHAVIOUR removeAnLoopRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveAnLoopRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.RemoveAnLoopRequestResult;
REGISTERED AS {q832-3Action 17};

removeAnLoopRequestBeh BEHAVIOUR
DEFINED AS
"This action is used by the OS of the SN to request the OS of the AN to remove a loop from a connection.";

5.5.18 removeLupsRequest (remove LUPs request)

removeLupsRequest ACTION
BEHAVIOUR removeLupsRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveLupsRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.RemoveLupsRequestResult;
REGISTERED AS {q832-3Action 18};

removeLupsRequestBeh BEHAVIOUR
DEFINED AS
"This action is used to request the peer OS to remove Logical User Ports from a VB5 interface.";

5.5.19 removeVb5ConnectionRequest (remove VB5 connection request)

removeVb5ConnectionRequest ACTION
BEHAVIOUR removeVb5ConnectionRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5ConnectionRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5ConnectionRequestResult;
REGISTERED AS {q832-3Action 19};

removeVb5ConnectionRequestBeh BEHAVIOUR

DEFINED AS

"This action is used to request the peer OS to remove a connection associated with a VB5 interface. The egress direction is out of the Access Network towards Service Node. The ingress direction is into the Access Network from the Service Node.";

5.5.20 removeVb5InterfaceRequest (remove VB5 interface request)

removeVb5InterfaceRequest ACTION

BEHAVIOUR removeVb5InterfaceRequestBeh;

MODE CONFIRMED;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5InterfaceRequestInfo;

WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5InterfaceRequestResult;

REGISTERED AS {q832-3Action 20};

removeVb5InterfaceRequestBeh BEHAVIOUR

DEFINED AS

"This action is used to request the peer OS to remove a VB5 interface.";

5.5.21 removeVb5ProtocolRequest (remove VB5 protocol request)

removeVb5ProtocolRequest ACTION

BEHAVIOUR removeVb5ProtocolRequestBeh;

MODE CONFIRMED;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5ProtocolRequestInfo;

WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5ProtocolRequestResult;

REGISTERED AS {q832-3Action 21};

removeVb5ProtocolRequestBeh BEHAVIOUR

DEFINED AS

"This action is used to request the peer OS to remove a protocol from a VB5 interface.";

5.5.22 removeVb5ProtocolVpRequest (remove VB5 protocol Vp request)

removeVb5ProtocolVpRequest ACTION

BEHAVIOUR removeVb5ProtocolVpRequestBeh;

MODE CONFIRMED;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5ProtocolVpRequestInfo;

WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5ProtocolVpRequestResult;

REGISTERED AS {q832-3Action 22};

removeVb5ProtocolVpRequestBeh BEHAVIOUR

DEFINED AS

"This action is used to request the peer OS to remove the protocol VP from a VB5 interface.";

5.5.23 removeVb5VcsRequest (remove VB5 VCs request)

removeVb5VcsRequest ACTION

BEHAVIOUR removeVb5VcsRequestBeh;

MODE CONFIRMED;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5VcsRequestInfo;

WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5VcsRequestResult;

REGISTERED AS {q832-3Action 23};

removeVb5VcsRequestBeh BEHAVIOUR

DEFINED AS

"This action is used to request the peer OS to remove VCs from a VP which is associated with a VB5 interface.";

5.5.24 removeVb5VpsRequest (remove VB5 VPs request)

removeVb5VpsRequest ACTION
BEHAVIOUR removeVb5VpsRequestBeh;
MODE CONFIRMED;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5VpsRequestInfo;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5VpsRequestResult;
REGISTERED AS {q832-3Action 24};

removeVb5VpsRequestBeh BEHAVIOUR
DEFINED AS
"This action is used to request the peer OS to remove VPs that are associated with a VB5 interface.";

5.5.25 snAccessLabelsInquiry (SN access labels inquiry)

snAccessLabelsInquiry ACTION
BEHAVIOUR snAccessLabelsInquiryBeh;
MODE CONFIRMED;
WITH REPLY SYNTAX Q832-3ASN1DefinedTypesModule.SnAccessLabelsInquiryResult;
REGISTERED AS {q832-3Action 25};

snAccessLabelsInquiryBeh BEHAVIOUR
DEFINED AS
"This action is used by the OS of an SN to inquire the access labels that an AN uses for the SN and the VB5 interface.";

5.6 Definición de notificaciones

5.6.1 addLupsIndication (add LUPs indication)

addLupsIndication NOTIFICATION
BEHAVIOUR addLupsIndicationBeh;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddLupsIndicationInfo;
REGISTERED AS {q832-3Notification 1};

addLupsIndicationBeh BEHAVIOUR
DEFINED AS
"This notification is used to notify the peer OS of the addition of Logical User Ports to a VB5 interface.";

5.6.2 addVb5ConnectionIndication (add VB5Connection indication)

addVb5ConnectionIndication NOTIFICATION
BEHAVIOUR addVb5ConnectionIndicationBeh;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5ConnectionIndicationInfo;
REGISTERED AS {q832-3Notification 2};

addVb5ConnectionIndicationBeh BEHAVIOUR
DEFINED AS
"This notification is used to notify the peer OS of the addition of a connection associated with a VB5 interface. The egress direction is out of the Access Network towards Service Node. The ingress direction is into the Access Network from the Service Node.";

5.6.3 addVb5InterfaceIndication (add VB5 interface indication)

addVb5InterfaceIndication NOTIFICATION
BEHAVIOUR addVb5InterfaceIndicationBeh;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5InterfaceIndicationInfo;
REGISTERED AS {q832-3Notification 3};

addVb5InterfaceIndicationBeh BEHAVIOUR
DEFINED AS

"This notification is used to inform the peer OS that a new VB5 interface has been added.";

5.6.4 addVb5ProtocolIndication (add VB5 protocol indication)

addVb5ProtocolIndication NOTIFICATION
BEHAVIOUR addVb5ProtocolIndicationBeh;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5ProtocolIndicationInfo;
REGISTERED AS {q832-3Notification 4};

addVb5ProtocolIndicationBeh BEHAVIOUR
DEFINED AS

"This notification is used to notify the peer OS of the addition of a protocol to an existing VB5 interface.";

5.6.5 addVb5ProtocolVpIndication (add VB5 protocol VP indication)

addVb5ProtocolVpIndication NOTIFICATION
BEHAVIOUR addVb5ProtocolVpIndicationBeh;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5ProtocolVpIndicationInfo;
REGISTERED AS {q832-3Notification 5};

addVb5ProtocolVpIndicationBeh BEHAVIOUR
DEFINED AS

"This notification is used to notify the peer OS of the addition of a protocol VP to a VB5 interface.";

5.6.6 addVb5VcsIndication (add VB5 VCs indication)

addVb5VcsIndication NOTIFICATION
BEHAVIOUR addVb5VcsIndicationBeh;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5VcsIndicationInfo;
REGISTERED AS {q832-3Notification 6};

addVb5VcsIndicationBeh BEHAVIOUR
DEFINED AS

"This notification is used to notify the peer OS of the addition of VCs to a VP which is associated with a VB5 interface.";

5.6.7 addVb5VpsIndication (add VB5 VPs indication)

addVb5VpsIndication NOTIFICATION
BEHAVIOUR addVb5VpsIndicationBeh;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.AddVb5VpsIndicationInfo;
REGISTERED AS {q832-3Notification 7};

addVb5VpsIndicationBeh BEHAVIOUR
DEFINED AS

"This notification is used to notify the peer OS of the addition of VPs that are associated with a VB5 interface.";

5.6.8 removeLupsIndication (remove LUPs indication)

removeLupsIndication NOTIFICATION
BEHAVIOUR removeLupsIndicationBeh;
WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveLupsIndicationInfo;
REGISTERED AS {q832-3Notification 8};

removeLupsIndicationBeh BEHAVIOUR
DEFINED AS

"This notification is used to notify the peer OS of the removal of Logical User Ports from a VB5 interface.";

5.6.9 removeVb5ConnectionIndication (remove VB5Connection indication)

removeVb5ConnectionIndication NOTIFICATION

BEHAVIOUR removeVb5ConnectionIndicationBeh;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5ConnectionIndicationInfo;

REGISTERED AS {q832-3Notification 9};

removeVb5ConnectionIndicationBeh BEHAVIOUR

DEFINED AS

"This notification is used to notify the peer OS of the removal of a connection associated with a VB5 interface. The egress direction is out of the Access Network towards Service Node. The ingress direction is into the Access Network from the Service Node.";

5.6.10 removeVb5InterfaceIndication (remove VB5 interface indication)

removeVb5InterfaceIndication NOTIFICATION

BEHAVIOUR removeVb5InterfaceIndicationBeh;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5InterfaceIndicationInfo;

REGISTERED AS {q832-3Notification 10};

removeVb5InterfaceIndicationBeh BEHAVIOUR

DEFINED AS

"This notification is used to notify the peer OS of the removal of a VB5 interface.";

5.6.11 removeVb5ProtocolIndication (remove VB5 protocol indication)

removeVb5ProtocolIndication NOTIFICATION

BEHAVIOUR removeVb5ProtocolIndicationBeh;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5ProtocolIndicationInfo;

REGISTERED AS {q832-3Notification 11};

removeVb5ProtocolIndicationBeh BEHAVIOUR

DEFINED AS

"This notification is used to notify the peer OS of the removal of a protocol from a VB5 interface.";

5.6.12 removeVb5ProtocolVpIndication (remove VB5 protocol VP indication)

removeVb5ProtocolVpIndication NOTIFICATION

BEHAVIOUR removeVb5ProtocolVpIndicationBeh;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5ProtocolVpIndicationInfo;

REGISTERED AS {q832-3Notification 12};

removeVb5ProtocolVpIndicationBeh BEHAVIOUR

DEFINED AS

"This notification is used to notify the peer OS of the removal of the protocol VP from a VB5 interface.";

5.6.13 removeVb5VcsIndication (remove VB5 VCs indication)

removeVb5VcsIndication NOTIFICATION

BEHAVIOUR removeVb5VcsIndicationBeh;

WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5VcsIndicationInfo;

REGISTERED AS {q832-3Notification 13};

removeVb5VcsIndicationBeh BEHAVIOUR

DEFINED AS

"This notification is used to notify the peer OS of the removal of VCs from a VP which is associated with a VB5 interface.";

5.6.14 removeVb5VpsIndication (remove VB5 VPs indication)

```
removeVb5VpsIndication NOTIFICATION
    BEHAVIOUR removeVb5VpsIndicationBeh;
    WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.RemoveVb5VpsIndicationInfo;
REGISTERED AS {q832-3Notification 14};

removeVb5VpsIndicationBeh BEHAVIOUR
    DEFINED AS
        "This notification is used to notify the peer OS of the removal of VPs that are associated with a VB5 interface.";
```

5.6.15 resourceStatusIndication (resource status indication)

```
resourceStatusIndication NOTIFICATION
    BEHAVIOUR resourceStatusIndicationBeh;
    WITH INFORMATION SYNTAX Q832-3ASN1DefinedTypesModule.ResourceStatusIndicationInfo;
REGISTERED AS {q832-3Notification 15};

resourceStatusIndicationBeh BEHAVIOUR
    DEFINED AS
        "This notification is used to notify the peer OS of the change of status of a Resource.";
```

6 Definiciones de tipos

```
Q832-3ASN1DefinedTypesModule {
    itu-t (0) recommendation (0) q (17) q832 (832) dot (127) coord (3)
        informationModel (0) asn1Modules (2) asn1DefinedTypesModule (0)}
```

DEFINITIONS IMPLICIT TAGS ::=

BEGIN

-- EXPORTS everything

IMPORTS

ObjectInstance

FROM CMIP-1 {joint-iso-itu-t ms (9) cmip (1) modules (0) protocol (3)}

NameType

FROM ASN1DefinedTypesModule {ccitt recommendation m 3100
informationModel(0) asn1Modules(2) asn1DefinedTypesModule(0)}

VciValue, VpiValue

FROM AtmMIBMod {itu-t (0) recommendation (0) i (9) atmm (751)
informationModel (0) asn1Module (2) atm(0)}

VpciValue

FROM AtmMIBMod {itu-t (0) recommendation (0) q (17) 824 (824) dot (127) bsm (6)
informationModel (0) asn1Modules (2) asn1DefinedTypesModule (0)}

; -- end of imports

-- start of object identifier definitions

q832-3InformationModel

OBJECT IDENTIFIER ::=

{itu-t (0) recommendation (0) q (17) q832 (832) dot (127) coord(3) informationModel(0)}

q832-3StandardSpecificExtension

OBJECT IDENTIFIER ::= {informationModel q832-3StandardSpecificExtension(0)}

q832-3ManagedObjectClass

OBJECT IDENTIFIER ::= {informationModel q832-3ManagedObjectClass(3)}

q832-3Package

OBJECT IDENTIFIER ::= {informationModel q832-3Package(4)}

q832-3NameBinding

OBJECT IDENTIFIER ::= {informationModel q832-3NameBinding(6)}

q832-3Attribute
 OBJECT IDENTIFIER ::= {informationModel q832-3Attribute (7)}
 q832-3Action
 OBJECT IDENTIFIER ::= {informationModel q832-3Action(9)}
 q832-3Notification
 OBJECT IDENTIFIER ::= {informationModel q832-3Notification(10)}

-- end of object identifier definitions

-- other ASN.1 definitions in alphabetical order – all these are new

AddAnLoopRequestInfo ::= SEQUENCE {
 logicalServicePortNumber[0] INTEGER,
 logicalUserPortNumber [1] INTEGER OPTIONAL,
 vpciValue [2] VpciValue,
 vciValue [3] VciValue OPTIONAL}

AddAnLoopRequestResult ::= CHOICE {
 loopAdded [0] NULL,
 loopNotAdded [1] LoopNotAddedInfo}

AddLupsIndicationInfo ::= AddLupsRequestInfo

AddLupsRequestInfo ::= SEQUENCE {
 logicalServicePortNumber[0] INTEGER,
 logicalUserPortNumber [1] SEQUENCE OF INTEGER}

AddLupsRequestResult ::= INTEGER {
 lupAdded (0),
 lupNotAdded (1)}

AddVb5ConnectionIndicationInfo ::= AddVb5ConnectionRequestInfo

AddVb5ConnectionRequestInfo ::= SEQUENCE {
 egressPeakCellRateCLP0Plus1 [0] INTEGER,
 egressPeakCellRateCLP0 [1] INTEGER,
 ingressPeakCellRateCLP0Plus1 [2] INTEGER,
 ingressPeakCellRateCLP0 [3] INTEGER,
 egressSustainableCellRateCLP0Plus1 [4] INTEGER,
 egressSustainableCellRateCLP0 [5] INTEGER,
 ingressSustainableCellRateCLP0Plus1 [6] INTEGER,
 ingressSustainableCellRateCLP0 [7] INTEGER,
 egressCDVToleranceCLP0Plus1 [8] INTEGER,
 egressCDVToleranceCLP0 [9] INTEGER,
 ingressCDVToleranceCLP0Plus1 [10] INTEGER,
 ingressCDVToleranceCLP0 [11] INTEGER,
 egressMaxBurstSizeCLP0Plus1 [12] INTEGER,
 egressMaxBurstSizeCLP0 [13] INTEGER,
 ingressMaxBurstSizeCLP0Plus1 [14] INTEGER,
 ingressMaxBurstSizeCLP0 [15] INTEGER,
 egressQosClass [16] INTEGER,
 ingressQosClass [17] INTEGER,
 vciValueA [18] INTEGER OPTIONAL,
 vciValueB [19] INTEGER OPTIONAL,
 physicalPortA [20] INTEGER,
 vpiValueA [21] INTEGER,
 logicalServicePortA [22] INTEGER OPTIONAL,
 vpciValueA [23] INTEGER OPTIONAL,
 physicalPortB [24] INTEGER,
 vpiValueB [25] INTEGER,
 logicalServicePortB [26] INTEGER OPTIONAL,
 vpciValueB [27] INTEGER OPTIONAL}

AddVb5ConnectionRequestResult ::= INTEGER {
 vb5ConnectionAdded **(0),**
 vb5ConnectionNotAdded **(1)}**

AddVb5InterfaceIndicationInfo ::= AddVb5InterfaceRequestInfo

AddVb5InterfaceRequestInfo ::= INTEGER -- *Logical service port number*

AddVb5InterfaceRequestResult ::= INTEGER {
 vb5InterfaceAdded **(0),**
 vb5InterfaceNotAdded **(1)}**

AddVb5ProtocolIndicationInfo ::= AddVb5ProtocolRequestInfo

AddVb5ProtocolRequestInfo ::= SEQUENCE {
 logicalServicePortNumber **[0] INTEGER,**
 vpciValue **[1] VpciValue,**
 maxNumVciBitsNearEnd **[2] INTEGER OPTIONAL,**
 maxNumVciBitsSupported **[3] INTEGER OPTIONAL,**
 maxNumActiveVccsAllowed **[4] INTEGER OPTIONAL,**
 maxNumActiveVccsNearEnd **[5] INTEGER OPTIONAL}**

AddVb5ProtocolRequestResult ::= INTEGER {
 vb5ProtocolAdded **(0),**
 vb5ProtocolNotAdded **(1)}**

AddVb5ProtocolVpIndicationInfo ::= AddVb5ProtocolVpRequestInfo

AddVb5ProtocolVpRequestInfo ::= SEQUENCE {
 logicalServicePortNumber**[0] INTEGER,**
 vpciValue **[1] VpciValue,**
 vpProfile **[2] VpProfile OPTIONAL}**

AddVb5ProtocolVpRequestResult ::= INTEGER {
 vb5ProtocolVpAdded **(0),**
 vb5ProtocolVpNotAdded **(1)}**

AddVb5VcsIndicationInfo ::= AddVb5VcsRequestInfo

AddVb5VcsRequestInfo ::= SEQUENCE {
 logicalServicePortNumber**[0] INTEGER,**
 logicalServiceSubport **[1] INTEGER OPTIONAL,**
 vpciValue **[2] VpciValue,**
 vciValue **[3] VciValue}**

AddVb5VcsRequestResult ::= INTEGER {
 vb5VcsAdded **(0),**
 vb5VcsNotAdded **(1)}**

AddVb5VpsIndicationInfo ::= AddVb5VpsRequestInfo

AddVb5VpsRequestInfo ::= SEQUENCE {
 logicalServicePortNumber**[0] INTEGER,**
 logicalServiceSubport **[1] INTEGER OPTIONAL,**
 addVpInfo **[2] AddVpInfo}**

AddVb5VpsRequestResult ::= INTEGER {
 vb5VpsAdded **(0),**
 vb5VpsNotAdded **(1)}**

AddVpInfo ::= SEQUENCE OF SEQUENCE {
 physicalPort **[0] INTEGER,**
 vpiValue **[1] VpiValue,**
 vpciValue **[2] VpciValue,**
vpProfile **[3] VpProfile OPTIONAL}**

AnServiceLabelInquiryResult ::= INTEGER OPTIONAL

AuditVb5ConnectionRequestInfo ::= SEQUENCE {
 logicalServicePortNumber**[0] INTEGER,**
 specifiedVpOrVc **[1] SpecifiedVpOrVc}**

AuditVb5ConnectionRequestResult ::= CHOICE {
 connectionAudited **[0] SpecifiedVpOrVc,**
 connectionNotAudited **[1] ConnectionNotAuditedInfo}**

AuditVb5VpciRequestInfo ::= SEQUENCE {
 logicalServicePortNumber **[0] INTEGER,**
 specifiedNniVpciOrRemoteVp **[1] SpecifiedNniVpciOrRemoteVp}**

AuditVb5VpciRequestResult ::= CHOICE {
 vpciAudited **[0] SpecifiedNniVpciOrRemoteVp,**
 vpciNotAudited **[1] VpciNotAuditedInfo}**

ConnectionNotAuditedInfo ::= INTEGER {
 unspecified **(0),**
 unknownPhysicalPort **(1),**
 unknownVpiValue **(2),**
 unknownVciValue **(3)}**

ListLupsRequestInfo ::= INTEGER *-- Logical service port number*

ListLupsFailureInfo ::= INTEGER {
 unspecified **(0),**
 unknownLspNumber **(1)}**

ListLupsRequestResult ::= CHOICE {
 success **[0] INTEGER, -- Logical user port number**
 failure **[1] ListLupsFailureInfo}**

ListVb5ProtocolDetailsRequestInfo ::= INTEGER *-- Logical service port number*

ListVb5ProtocolDetailsRequestResult ::= CHOICE {
 success **[0] ListVb5ProtocolDetailsSuccessInfo,**
 failure **[1] ListVb5ProtocolDetailsFailureInfo}**

ListVb5ProtocolDetailsFailureInfo ::= INTEGER {
 unspecified **(0),**
 unknownLspNumber **(1)}**

ListVb5ProtocolDetailsSuccessInfo ::= SEQUENCE OF Vb5ProtocolDetails

Vb5ProtocolDetails ::= SEQUENCE {
 protocolType **[0] ProtocolType,**
 vpciValue **[1] VpciValue,**
 vciValue **[2] VciValue,**
 egressPeakCellRateCLP0Plus1 **[3] INTEGER,**
 egressPeakCellRateCLP0 **[4] INTEGER,**
 ingressPeakCellRateCLP0Plus1 **[5] INTEGER,**
 ingressPeakCellRateCLP0 **[6] INTEGER,**
 egressSustainableCellRateCLP0Plus1 **[7] INTEGER,**
 egressSustainableCellRateCLP0 **[8] INTEGER,**
 ingressSustainableCellRateCLP0Plus1 **[9] INTEGER,**

ingressSustainableCellRateCLP0	[10]	INTEGER,
egressCDVToleranceCLP0Plus1	[11]	INTEGER,
egressCDVToleranceCLP0	[12]	INTEGER,
ingressCDVToleranceCLP0Plus1	[13]	INTEGER,
ingressCDVToleranceCLP0	[14]	INTEGER,
egressMaxBurstSizeCLP0Plus1	[15]	INTEGER,
egressMaxBurstSizeCLP0	[16]	INTEGER,
ingressMaxBurstSizeCLP0Plus1	[17]	INTEGER,
ingressMaxBurstSizeCLP0	[18]	INTEGER,
bufferRelease	[19]	BOOLEAN,
maxCc	[20]	INTEGER,
maxInformationFieldLength	[21]	INTEGER,
maxLengthSscopUuField	[22]	INTEGER,
maxPd	[23]	INTEGER,
maxSscopCreditToPeer	[24]	INTEGER,
maxStat	[25]	INTEGER,
sscopTimerCc	[26]	INTEGER,
sscopTimerIdle	[27]	INTEGER,
sscopTimerKeepAlive	[28]	INTEGER,
sscopTimerNoResponse	[29]	INTEGER,
sscopTimerPoll	[30]	INTEGER}

ProtocolType ::= INTEGER {
RTMC (1),
BBCC (2)}

ListVb5InterfacesRequestInfo ::= SEQUENCE OF ObjectInstance

ListVb5InterfacesRequestResult ::= SEQUENCE OF INTEGER -- Logical service port number

ListVcsFailureInfo ::= INTEGER {
unspecified (0),
unknownLspNumber (1),
unknownLupNumber (2),
unknownVpciValue (3),
unknownLupVpciCombination (4)}

ListVb5VcsRequestInfo ::= SEQUENCE {
logicalServicePortNumber [0] INTEGER,
logicalUserPortNumber [1] INTEGER OPTIONAL,
vpciValue [2] VpciValue OPTIONAL}

ListVb5VcsRequestResult ::= CHOICE {
success [0] ListVcsSuccessInfo,
failure [1] ListVcsFailureInfo}

ListVcsSuccessInfo ::= SEQUENCE {
logicalUserPortNumber [0] INTEGER OPTIONAL,
vpciValue [1] VpciValue,
vciValue [2] VciValue}

ListVb5VpsRequestInfo ::= SEQUENCE {
logicalServicePortNumber [0] INTEGER,
logicalUserPortNumber [1] INTEGER OPTIONAL}

ListVb5VpsRequestResult ::= CHOICE {
success [0] ListVb5VpsSuccessInfo,
failure [1] ListVb5VpFailureInfo}

ListVb5VpFailureInfo ::= INTEGER {
unspecified (0),
unknownLspNumber (1),

unknownLupNumber (2)}

ListVb5VpsSuccessInfo ::= SEQUENCE OF Vb5VpDetails

Vb5VpDetails ::= SEQUENCE {
 logicalUserPortNumber [0] INTEGER OPTIONAL,
 physicalPort [1] INTEGER,
 vpiValue [2] INTEGER,
 vpciValue [3] INTEGER OPTIONAL,
 maxNumVciBitsNearEnd [4] INTEGER OPTIONAL,
 maxNumVciBitsSupported [5] INTEGER OPTIONAL,
 maxNumActiveVccsAllowed [6] INTEGER OPTIONAL,
 maxNumActiveVccsNearEnd [7] INTEGER OPTIONAL}

LoopNotAddedInfo ::= INTEGER {
 unspecified (0),
 unknownLogicalServicePort (1),
 unknownLogicalUserPort (2),
 unknownVpciValue (3),
 unknownVciValue (4),
 loopAlreadyPresent (5)}

LoopNotRemovedInfo ::= INTEGER {
 unspecified (0),
 unknownLogicalServicePort (1),
 unknownLogicalUserPort (2),
 unknownVpciValue (3),
 unknownVciValue (4),
 noLoopPresent (5)}

RemoveAnLoopRequestInfo ::= AddAnLoopRequestInfo

RemoveAnLoopRequestResult ::= CHOICE {
 loopRemoved [0] NULL,
 loopNotRemoved [1] LoopNotRemovedInfo}

RemoveLupsIndicationInfo ::= RemoveLupsRequestInfo

RemoveLupsRequestInfo ::= AddLupsRequestInfo

RemoveLupsRequestResult ::= INTEGER {
 lupRemoved (0),
 lupNotRemoved (1)}

RemoveVb5ConnectionIndicationInfo ::= RemoveVb5ConnectionRequestInfo

RemoveVb5ConnectionRequestInfo ::= SEQUENCE {
 vciValueA [0] INTEGER OPTIONAL,
 vciValueB [1] INTEGER OPTIONAL,
 physicalPortA [2] INTEGER,
 vpiValueA [3] INTEGER,
 logicalServicePortA [4] INTEGER OPTIONAL,
 vpciValueA [5] INTEGER OPTIONAL,
 physicalPortB [6] INTEGER,
 vpiValueB [7] INTEGER,
 logicalServicePortB [8] INTEGER OPTIONAL,
 vpciValueB [9] INTEGER OPTIONAL}

RemoveVb5ConnectionRequestResult ::= INTEGER {
 vb5ConnectionRemoved (0),
 vb5ConnectionNotRemoved (1)}

fault (3)}

SnAccessLabelsInquiryResult ::= SEQUENCE {
 snLabel [0] INTEGER OPTIONAL,
 interfaceLabel [1] INTEGER OPTIONAL}

SpecifiedNniVpci ::= SEQUENCE {
 logicalUserPortNumber [0] INTEGER OPTIONAL,
 vpciValue [1] VpciValue}

SpecifiedNniVpciOrRemoteVp ::= CHOICE {
 specifiedNniVpci [0] SpecifiedNniVpci,
 specifiedRemoteVp [1] SpecifiedVp}

SpecifiedVc ::= SEQUENCE {
 physicalPort [0] INTEGER,
 vpiValue [1] VpiValue,
 vciValue [2] VciValue}

SpecifiedVp ::= SEQUENCE {
 physicalPort [0] INTEGER,
 vpiValue [1] VpiValue}

SpecifiedVpOrVc ::= CHOICE {
 specifiedVp [0] SpecifiedVp,
 specifiedVc [1] SpecifiedVc}

Vb5ProtocolType ::= INTEGER {
 rtmc (0),
 bbcc (1)}

VpciNotAuditedInfo ::= INTEGER {
 unspecified (0),
 unknownLupNumber (1),
 unknownVpci (2),
 unknownPhysicalPort (3),
 unknownVpiValue (4)}

VpProfile ::= SEQUENCE {
 maxNumVciBitsNearEnd [0] INTEGER,
 maxNumVciBitsSupported [1] INTEGER,
 maxNumActiveVccsAllowed [2] INTEGER,
 maxNumActiveVccsNearEnd [3] INTEGER}

END – of Q832-3ASN1DefinedTypesModule

7 Pilas de protocolos

Las pilas de protocolos especificadas en UIT-T Q.811, Q.812, G.773 y la parte transconexiones digitales SDH de UIT-T G.784 pueden utilizarse como parte de la pila de protocolos para esta Recomendación. Deberán utilizarse las siguientes Recomendaciones para ampliar estas pilas de protocolos de modo que incluyan ATM:

- Q.2811 Interfaces Q3 y X de banda ancha – Protocolos de capa inferior
- Q.2812 Interfaces Q3 y X de banda ancha – Protocolos de capa superior

ANEXO A

Necesidades de gestión

Las necesidades de gestión se indican a continuación y en los principios para la provisión de las interfaces VB5.1 y VB5.2.

A.1 Necesidades de gestión de carácter general

Incluyen las funciones generales de coordinación de la gestión entre la red de acceso y el nodo de servicio a través de las interfaces Q3/X.

A.1.1 Coordinación de operaciones relativas a la configuración de los VP y VC

La función de gestión de la configuración deberá soportar la adición y supresión coordinadas de VP y VC tanto en las interfaces UNI como en las interfaces VB5, de manera que se pueda añadir y suprimir VP y VC sin sufrir trastornos.

A.1.2 Comprobación de la VPC

Se requiere un mecanismo para comprobar la identidad de las VPC que se establecen entre un puerto de usuario y un nodo de servicio, de modo que las equivocaciones que se produzcan en la conexión cruzada dentro de una red acceso puedan ser identificadas.

A.1.3 Coordinación de datos de configuración de puerto

La coordinación de la información de configuración relativa a los puertos de usuario y los puertos de servicio y sus VP y VC es necesaria para asegurar la coherencia entre la red de acceso y el nodo de servicio.

A.1.4 Coordinación de valores de VPCI

Se requiere la coordinación de la gestión entre el SN y la AN en lo que respecta a la asignación de valores de VPCI para conexiones.

A.1.5 Coherencia de la configuración

Es necesario comprobar la coherencia de la configuración de puertos de usuario lógicos, puertos de servicio lógicos, y accesos UNI.

A.1.6 Disponibilidad de la información

La información sobre la interfaz VB5 debe ser visible solamente por los operadores que están relacionados con esa interfaz VB5.

A.2 Coordinador de la interfaz VB5

A.2.1 Creación

Es necesario coordinar la creación de la interfaz VB5, utilizando la siguiente información:

- 1) identificadores de VP;
- 2) VCC asignadas para protocolos B-BCC y RTMC;
- 3) gama de VCI;
- 4) número máximo de VCC activas simultáneamente;

- 5) anchura de banda máxima de la VPC en ambos sentidos (egreso e ingreso), proporcionada por el descriptor de tráfico, que especifica, por ejemplo:
 - velocidad máxima de cresta;
 - tolerancia a la variación del retardo de célula;
 - velocidad de célula sostenible;
 - tamaño máximo de ráfaga;
- 6) clase de calidad de servicio del VP;
- 7) otros VC provistos en el VP;
- 8) el perfil de tráfico del VC, que describe, por ejemplo:
 - velocidad máxima de cresta;
 - tolerancia a la variación del retardo de célula;
 - velocidad de célula sostenible;
 - tamaño máximo de ráfaga;
- 9) clase de calidad de servicio del VC.

A.2.2 Verificación y auditoría

Es necesario verificar y auditar la configuración correcta de la interfaz VB5 (para la correspondiente información, véase la sección relativa a la creación).

A.2.3 Modificación

Es necesario coordinar la modificación de la interfaz VB5.

A.2.4 Supresión

Es necesario coordinar la supresión de la interfaz VB5.

A.2.5 Provisión de VC y VP

Es necesario coordinar la provisión de VC y VP, a los usuarios, utilizando la siguiente información:

- 1) identificadores de VP/VC;
- 2) gama de VCI para VP;
- 3) número máximo de VCC activas simultáneamente en VP;
- 4) anchura de banda máxima de la VPC en ambos sentidos (egreso e ingreso), proporcionada por el descriptor de tráfico, que especifica, por ejemplo:
 - velocidad máxima de cresta;
 - tolerancia a la variación del retardo de célula;
 - velocidad de célula sostenible;
 - tamaño máximo de ráfaga;
- 5) clase de calidad de servicio del VP;
- 6) VC provistos en el VP;
- 7) el perfil de tráfico del VC, que describe, por ejemplo:
 - velocidad máxima de cresta;
 - tolerancia a la variación del retardo de célula;
 - velocidad de célula sostenible;
 - tamaño máximo de ráfaga.

A.2.6 Verificación y auditoría

Es necesario verificar y auditorizar la configuración correcta de VC y VP para usuarios (para la correspondiente información, véase la sección relativa a la provisión).

A.2.7 Modificación de VP y VC

Es necesario coordinar la modificación de VC y VP para usuarios.

A.2.8 Verificación y auditoría

Es necesario verificar y auditorizar la configuración correcta de la interfaz VB5 (para la correspondiente información, véase la sección relativa a la creación).

A.2.9 Supresión de VP y VC

Es necesario coordinar la supresión de VC y VP para usuarios.

A.3 Coordinación de la interfaz UNI

A.3.1 Creación

Es necesario coordinar la creación de VP de usuario y de VC de usuario entre los usuarios y la AN, utilizando la siguiente información:

- 1) identificadores de VP/VC;
- 2) gama de VCI;
- 3) número máximo de VCC activas simultáneamente;
- 4) anchura de banda máxima de la VPC en ambos sentidos (egreso e ingreso), proporcionada por el descriptor de tráfico, que especifica, por ejemplo:
 - velocidad máxima de cresta;
 - tolerancia a la variación del retardo de célula;
 - velocidad de célula sostenible;
 - tamaño máximo de ráfaga;
- 5) clase de calidad de servicio del VP;
- 6) el perfil de tráfico del VC, que describe, por ejemplo:
 - velocidad máxima de cresta;
 - tolerancia a la variación del retardo de célula;
 - velocidad de célula sostenible;
 - tamaño máximo de ráfaga;
- 7) clase de calidad de servicio del VC.

A.3.2 Verificación y auditoría

Es necesario verificar y auditorizar la configuración correcta de VC y VP entre los usuarios y la AN (para la correspondiente información, véase la sección relativa a la creación).

A.3.3 Modificación

Es necesario coordinar la modificación de VP de usuario y VC de usuario entre los usuarios y la AN.

A.3.4 Supresión

Es necesario coordinar la supresión de VP de usuario y VC de usuario entre los usuarios y la AN.

A.4 Necesidades de coordinación del control de conexión portadora de banda ancha

Esta coordinación debe efectuarse porque es necesario soportar el protocolo VB5.2 B-BCC [3].

La VPC/VCC en la interfaz VB5 y en la interfaz UNI puede ser seleccionada por el SN y por la AN; en consecuencia, tanto uno como la otra deben disponer de la información adecuada para realizar las funciones de control de admisión de conexión (CAC, *connection admission control*) y seleccionar los identificadores de VPC/VC correctos.

A.4.1 VC en la interfaz VB5

Para el soporte de la creación, supresión y modificación de un VC en la interfaz VB5.2 de acuerdo con los procedimientos de B-BCC se debe disponer de la siguiente información en la AN y el SN:

- las VPC asociadas a la interfaz VB5.2; la información asociada a cada VPC incluirá:
 - 1) gama de VCI;
 - 2) número máximo de VCC activas simultáneamente;
 - 3) anchura de banda máxima en ambos sentidos (egreso e ingreso) de la VPC, proporcionada por el descriptor de tráfico, que especifica, por ejemplo:
 - velocidad máxima de cresta;
 - tolerancia a la variación del retardo de célula;
 - velocidad de célula sostenible;
 - tamaño máximo de ráfaga;
 - 4) clase de calidad de servicio del VP;
 - 5) VC asignados en el VP;
 - 6) el perfil de tráfico del VC, que describe, por ejemplo:
 - velocidad máxima de cresta;
 - tolerancia a la variación del retardo de célula;
 - velocidad de célula sostenible;
 - tamaño máximo de ráfaga;
 - 7) clase de calidad de servicio del VC.

Esta información está disponible en la AN y el SN, y la proporcionan los puntos de conexión de VP/VC que modelan las clases de objeto gestionado, definidos en UIT-T I.751.

A.4.2 VC en la interfaz UNI

Para el soporte de la creación, supresión y modificación de un VC en la interfaz UNI de acuerdo con los procedimientos de B-BCC se debe disponer de la siguiente información en la AN y el SN:

- las VPC asociadas al LUP; la información asociada a cada VPC incluirá:
 - 1) gama de VCI;
 - 2) número máximo de VCC activas simultáneamente;
 - 3) anchura de banda máxima de la VPC en ambos sentidos (egreso e ingreso), proporcionada por el descriptor de tráfico, que especifica, por ejemplo:
 - velocidad máxima de cresta;
 - tolerancia a la variación del retardo de célula;
 - velocidad de célula sostenible;
 - tamaño máximo de ráfaga;
 - 4) clase de calidad de servicio del VP;

- 5) VC asignados en el VP;
- 6) el perfil de tráfico del VC, que describe, por ejemplo:
 - velocidad máxima de cresta;
 - tolerancia a la variación del retardo de célula;
 - velocidad de célula sostenible;
 - tamaño máximo de ráfaga;
- 7) clase de calidad de servicio del VC.

Esta información está disponible en la AN y la proporcionan los puntos de conexión de VP/VC que modelan las clases de objeto gestionado, definidos en UIT-T I.751. Se proporcionará la misma información al SN.

A.5 Gestión de las averías y de la calidad de funcionamiento

En lo que respecta a la gestión de las averías y la calidad de funcionamiento se han señalado las siguientes necesidades.

A.5.1 Informes de averías

Las averías deberán ser informadas a través de la interfaz X cuando las interfaces de tráfico VB5 no están en condiciones de funcionamiento.

A.5.2 Localización de averías

La interfaz X soportará las actividades de coordinación que permiten a un nodo de servicio y a una red de acceso cooperar en la localización del origen de las averías, por ejemplo cuando se utilizan conexiones en bucle.

ANEXO B

Necesidades de transacciones

Las transacciones presentadas a continuación están destinadas a ser incorporadas en protocolos de implementación. Se han definido utilizando la especificación de invocación de procedimiento distante [7], pues esta norma proporciona un método simple para la especificación formal de las transacciones, que es compatible con el protocolo CMIP, pero no está limitado al mismo. Este anexo no especifica una implementación obligatoria del RPC porque las definiciones de parámetros no están consolidadas, ya que el objetivo que se persigue es la claridad, no la eficiencia; esto, sin embargo, no excluye que este anexo se utilice como una implementación del RPC.

```

/*****
*****

```

*** Transaction Definitions**

```

*****
*****/

```

```

program XVB5 {
version {
/* */
addAnLoopRequestAnswer AddAnLoopRequest (addAnLoopRequestCall) = 1
addLupsIndicationAnswer AddLupsIndication (addLupsIndicationCall) = 2
addLupsRequestAnswer AddLupsRequest (addLupsRequestCall) = 3
addVb5ProtocolVpIndicationAnswer AddVb5ProtocolVpIndication (addVb5ProtocolVpIndicationCall) = 4
addVb5ProtocolVpRequestAnswer AddVb5ProtocolVpRequest (addVb5ProtocolVpRequestCall) = 5

```

```

addVb5InterfaceRequestAnswer AddVb5InterfaceRequest (addVb5InterfaceRequestCall) = 6
addVb5InterfaceIndicationAnswer AddVb5InterfaceIndication (addVb5InterfaceIndicationCall) = 7
addVb5ProtocolIndicationAnswer AddVb5ProtocolIndication (addVb5ProtocolIndicationCall) = 8
addVb5ProtocolRequestAnswer AddVb5ProtocolRequest (addVb5ProtocolRequestCall) = 9
addVb5VcsIndicationAnswer AddVb5VcsIndication (addVb5VcsIndicationCall) = 10
addVb5VcsRequestAnswer AddVb5VcsRequest (addVb5VcsRequestCall) = 11
addVb5VpsIndicationAnswer AddVb5VpsIndication (addVb5VpsIndicationCall) = 12
addVb5VpsRequestAnswer AddVb5VpsRequest (addVb5VpsRequestCall) = 13
addVb5ConnectionIndicationAnswer AddVb5ConnectionIndication (addVb5ConnectionIndicationCall) = 14
addVb5ConnectionRequestAnswer AddVb5ConnectionRequest (addVb5ConnectionRequestCall) = 15
auditVb5ConnectionRequestAnswer AuditVb5ConnectionRequest (auditVb5ConnectionRequestCall) = 16
auditVb5VpciRequestAnswer AuditVb5VpciRequest (auditVb5VpciRequestCall) = 17
listLupsRequestAnswer ListLupsRequest (listLupsRequestCall) = 18
listVb5ProtocolDetailsRequestAnswer ListVb5ProtocolDetailsRequest (listVb5ProtocolDetailsRequestCall) = 19
listVb5InterfacesRequestAnswer ListVb5InterfacesRequest (listVb5InterfacesRequestCall) = 20
listVb5VcsRequestAnswer ListVb5VcsRequest (listVb5VcsRequestCall) = 21
listVb5VpsRequestAnswer ListVb5VpsRequest (listVb5VpsRequestCall) = 22
removeAnLoopRequestAnswer RemoveAnLoopRequest (removeAnLoopRequestCall) = 23
removeLupsIndicationAnswer RemoveLupsIndication (removeLupsIndicationCall) = 24
removeLupsRequestAnswer RemoveLupsRequest (removeLupsRequestCall) = 25
removeVb5ProtocolVpIndicationAnswer RemoveVb5ProtocolVpIndication
(removeVb5ProtocolVpIndicationCall) = 26
removeVb5ProtocolVpRequestAnswer RemoveVb5ProtocolVpRequest (removeVb5ProtocolVpRequestCall) = 27
removeVb5InterfaceIndicationAnswer RemoveVb5InterfaceIndication (removeVb5InterfaceIndicationCall) = 28
removeVb5InterfaceRequestAnswer RemoveVb5InterfaceRequest (removeVb5InterfaceRequestCall) = 29
removeVb5ProtocolIndicationAnswer RemoveVb5ProtocolIndication (removeVb5ProtocolIndicationCall) = 30
removeVb5ProtocolRequestAnswer RemoveVb5ProtocolRequest (removeVb5ProtocolRequestCall) = 31
removeVb5VcsIndicationAnswer RemoveVb5VcsIndication (removeVb5VcsIndicationCall) = 32
removeVb5VcsRequestAnswer RemoveVb5VcsRequest (removeVb5VcsRequestCall) = 33
removeVb5VpsIndicationAnswer RemoveVb5VpsIndication (removeVb5VpsIndicationCall) = 34
removeVb5VpsRequestAnswer RemoveVb5VpsRequest (removeVb5VpsRequestCall) = 35
removeVb5ConnectionIndicationAnswer RemoveVb5ConnectionIndication
(removeVb5ConnectionIndicationCall) = 36
removeVb5ConnectionRequestAnswer RemoveVb5ConnectionRequest (removeVb5ConnectionRequestCall) = 37
anServiceLabelInquiryAnswer AnServiceLabelInquiry (anServiceLabelInquiryCall) = 38
snAccessLabelsInquiryAnswer SnAccessLabelsInquiry (snAccessLabelsInquiryCall) = 39
resourceStatusIndicationAnswer ResourceStatusIndication (resourceStatusIndicationCall) = 40
/* */
} = 1 /* denotes version 1 */
} = 8323 /* denotes Q.832.3 */

```

```

/*****
****

```

*** Parameter Definitions**

```

*****
****

```

```

/*****
****

```

*** Parameter Definitions for Transaction 1**

```

*****

```

This transaction is used by the OS of the SN to request the OS of the AN to loop a connection so that cells sent to the AN will be returned.

```

*/

```

```

struct addAnLoopRequestCall {
    unsigned int LogicalServicePort;
    union switch (bool VpcReachesUserPort) {
        case TRUE:
            unsigned int LogicalUserPort;
        case FALSE:
            void;
    } OptionalUserPort;
    unsigned int VpciValue;
}

```

```

        union switch (bool VccLoopback) {
            case TRUE:
                unsigned int VciValue;
            case FALSE:
                void;
        } OptionalVciValue;
};
struct addAnLoopRequestAnswer {
    union switch (bool Failure) {
        case TRUE:
            enum {
                Unspecified                = 0,
                UnknownLogicalServicePort  = 1,
                UnknownLogicalUserPort     = 2,
                UnknownVpciValue           = 3,
                UnknownVciValue            = 4,
                LoopAlreadyPresent          = 5
            } FailureReason;
        case FALSE:
            void;
    } OptionalFailureReason;
};

/*****
* Parameter Definitions for Transaction 2
*****/
This transaction is used to notify the peer OS of the addition of logical user ports to a VB5 interface.
*/
struct addLupsIndicationCall {
    unsigned int LogicalServicePortNumber;
    unsigned int LogicalUserPortNumber<>;
};
struct addLupsIndicationAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 3
*****/
This transaction is used to request the peer OS to add logical user ports to a VB5 interface.
*/
struct addLupsRequestCall {
    unsigned int LogicalServicePortNumber;
    unsigned int LogicalUserPortNumber<>;
};
struct addLupsRequestAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 4
*****/
This transaction is used to notify the peer OS of the addition of a protocol VP to a VB5 interface.
*/
struct addVb5ProtocolVpIndicationCall {
    unsigned int LogicalServicePortNumber;
    unsigned int VpciValue;
    union switch (bool ProfileSupported) {
        case TRUE:
            struct {
                unsigned int maxNumVciBitsNearEnd;
                unsigned int maxNumVciBitsSupported;
                unsigned int maxNumActiveVccsAllowed;
            }
    }
};

```

```

        unsigned int maxNumActiveVccsNearEnd;
    } Profile;
case FALSE:
    void;
} OptionalProfile
};
struct addVb5ProtocolVpIndicationAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 5
*****/
This transaction is used to request the peer OS to add a protocol VP to a VB5 interface.
*/
struct addVb5ProtocolVpRequestCall {
    unsigned int LogicalServicePortNumber;
    unsigned int VpciValue;
    union switch (bool ProfileSupported) {
        case TRUE:
            struct {
                unsigned int maxNumVciBitsNearEnd;
                unsigned int maxNumVciBitsSupported;
                unsigned int maxNumActiveVccsAllowed;
                unsigned int maxNumActiveVccsNearEnd;
            } Profile;
        case FALSE:
            void;
    } OptionalProfile
};
struct addVb5ProtocolVpRequestAnswer {
    bool Acknowledge;
};

/*****
Parameter Definitions for Transaction 6
*****/
This transaction is used to inform the peer OS that a new VB5 interface has been added.
*/
struct addVb5InterfaceIndicationCall {
    unsigned int LogicalServicePortNumber;
};
struct addVb5InterfaceIndicationAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 7
*****/
This transaction is used to request the peer OS to add a VB5 interface.
*/
struct addVb5InterfaceRequestCall {
    unsigned int LogicalServicePortNumber;
};
struct addVb5InterfaceRequestAnswer {
    bool Acknowledge;
};

```

/******

* Parameter Definitions for Transaction 8

This transaction is used to notify the peer OS of the addition of a protocol to an existing VB5 interface.

*/

```
struct addVb5ProtocolIndicationCall {
    unsigned int LogicalServicePortNumber;
    unsigned int VpciValue;
    unsigned int VciValue;
    enum {
        RTMC      = 1,
        BBCC      = 2
    } ProtocolType;
    struct {
        /* Cell Rate is in cells per second. */
        /* Cell Delay Variation is in microseconds. */
        /* Burst Size is in cells. */
        unsigned int EgressPeakCellRateCLP0Plus1;
        unsigned int EgressPeakCellRateCLP0;
        unsigned int IngressPeakCellRateCLP0Plus1;
        unsigned int IngressPeakCellRateCLP0;
        unsigned int EgressSustainableCellRateCLP0Plus1;
        unsigned int EgressSustainableCellRateCLP0;
        unsigned int IngressSustainableCellRateCLP0Plus1;
        unsigned int IngressSustainableCellRateCLP0;
        unsigned int EgressCDVToleranceCLP0Plus1;
        unsigned int EgressCDVToleranceCLP0;
        unsigned int IngressCDVToleranceCLP0Plus1;
        unsigned int IngressCDVToleranceCLP0;
        unsigned int EgressMaxBurstSizeCLP0Plus1;
        unsigned int EgressMaxBurstSizeCLP0;
        unsigned int IngressMaxBurstSizeCLP0Plus1;
        unsigned int IngressMaxBurstSizeCLP0;
    } QosParameters;
    struct {
        bool BufferRelease;      /* on connection release */
        unsigned int MaxCc;      /* maximum value in PDUs of the state variable VT(CC) */
        unsigned int MaxInformationFieldLength; /* in octets */
        unsigned int MaxLengthSscopUuField; /* in octets */
        unsigned int MaxPd;      /* maximum value in PDUs of the state variable VT(PD) */
        unsigned int MaxSscopCreditToPeer; /* absolute value in PDUs of the receive window */
        unsigned int MaxStat; /* maximum number of list elements in a STAT PDU */
        unsigned int SscopTimerCc; /* time interval between transmissions in milliseconds */
        unsigned int SscopTimerIdle; /* idle phase in milliseconds */
        unsigned int SscopTimerKeepAlive; /* keep alive phase in milliseconds */
        unsigned int SscopTimerNoResponse; /* no response time in milliseconds */
        unsigned int SscopTimerPoll; /* active POLL phase in milliseconds */
    } SaalParameters;
};

struct addVb5ProtocolIndicationAnswer {
    bool Acknowledge;
};
```

/******

* Parameter Definitions for Transaction 9

This transaction is used to request the peer OS to add a protocol to an existing VB5 interface.

*/

```
struct addVb5ProtocolRequestCall {
    unsigned int LogicalServicePortNumber;
    unsigned int VpciValue;
    unsigned int VciValue;
    enum {
```



```

        RTMC      = 1,
        BBCC      = 2
    } ProtocolType;
    struct {
        /* Cell Rate is in cells per second. */
        /* Cell Delay Variation is in microseconds. */
        /* Burst Size is in cells. */
        unsigned int EgressPeakCellRateCLP0Plus1;
        unsigned int EgressPeakCellRateCLP0;
        unsigned int IngressPeakCellRateCLP0Plus1;
        unsigned int IngressPeakCellRateCLP0;
        unsigned int EgressSustainableCellRateCLP0Plus1;
        unsigned int EgressSustainableCellRateCLP0;
        unsigned int IngressSustainableCellRateCLP0Plus1;
        unsigned int IngressSustainableCellRateCLP0;
        unsigned int EgressCDVToleranceCLP0Plus1;
        unsigned int EgressCDVToleranceCLP0;
        unsigned int IngressCDVToleranceCLP0Plus1;
        unsigned int IngressCDVToleranceCLP0;
        unsigned int EgressMaxBurstSizeCLP0Plus1;
        unsigned int EgressMaxBurstSizeCLP0;
        unsigned int IngressMaxBurstSizeCLP0Plus1;
        unsigned int IngressMaxBurstSizeCLP0;
    } QosParameters;
    struct {
        bool BufferRelease;      /* on connection release */
        unsigned int MaxCc;      /* maximum value in PDUs of the state variable VT(CC) */
        unsigned int MaxInformationFieldLength; /* in octets */
        unsigned int MaxLengthSscopUuField; /* in octets */
        unsigned int MaxPd;      /* maximum value in PDUs of the state variable VT(PD) */
        unsigned int MaxSscopCreditToPeer; /* absolute value in PDUs of the receive window */
        unsigned int MaxStat;     /* maximum number of list elements in a STAT PDU */
        unsigned int SscopTimerCc; /* time interval between transmissions in milliseconds */
        unsigned int SscopTimerIdle; /* idle phase in milliseconds */
        unsigned int SscopTimerKeepAlive; /* keep alive phase in milliseconds */
        unsigned int SscopTimerNoResponse; /* no response time in milliseconds */
        unsigned int SscopTimerPoll; /* active POLL phase in milliseconds */
    } SaalParameters;
};
struct addVb5ProtocolRequestAnswer {
    bool Acknowledge;

    /**
     * Parameter Definitions for Transaction 10
     */
    This transaction is used to notify the peer OS of the addition of VCs to a VP which is associated with a VB5
    interface.
    */
    struct addVb5VcsIndicationCall {
        unsigned int LogicalServicePortNumber;
        unsigned int *LogicalServiceSubport;
        /* present if VPC reaches a user port */
        unsigned int VpciValue;
        unsigned int VciValue<>;
    };
};
struct addVb5VcsIndicationAnswer {
    bool Acknowledge;
};

```

/******

*** Parameter Definitions for Transaction 11**

This transaction is used to request the peer OS to add VCs to a VP which is associated with a VB5 interface.

*/

```
struct addVb5VcsRequestCall {
    unsigned int LogicalServicePortNumber;
    unsigned int *LogicalServiceSubport;
    /* present if VPC reaches a user port */
    unsigned int VpciValue;
    unsigned int VciValue<>;
};
struct addVb5VcsRequestAnswer {
    bool Acknowledge;
};
```

/******

*** Parameter Definitions for Transaction 12**

This transaction is used to notify the peer OS of the addition of VPs that are associated with a VB5 interface.

*/

```
struct addVb5VpsIndicationCall {
    unsigned int LogicalServicePortNumber;
    unsigned int *LogicalServiceSubport;
    /* present if VPC reaches a user port */
    struct {
        unsigned int PhysicalPort;
        unsigned int VpiValue;
        unsigned int VpciValue;
        union switch (bool ProfileSupported) {
            case TRUE:
                struct {
                    unsigned int maxNumVciBitsNearEnd;
                    unsigned int maxNumVciBitsSupported;
                    unsigned int maxNumActiveVccsAllowed;
                    unsigned int maxNumActiveVccsNearEnd;
                } Profile;
            case FALSE:
                void;
        } OptionalProfile
    } VpInfo<>;
};
struct addVb5VpsIndicationAnswer {
    bool Acknowledge;
};
```

/******

*** Parameter Definitions for Transaction 13**

This transaction is used to request the peer OS to add VPs that are associated with a VB5 interface.

*/

```
struct addVb5VpsRequestCall {
    unsigned int LogicalServicePortNumber;
    unsigned int *LogicalServiceSubport;
    /* present if VPC reaches a user port */
    struct {
        unsigned int PhysicalPort;
        unsigned int VpiValue;
        unsigned int VpciValue;
        union switch (bool ProfileSupported) {
            case TRUE:
                struct {
                    unsigned int maxNumVciBitsNearEnd;
                } Profile;
            case FALSE:
                void;
        } OptionalProfile
    } VpInfo<>;
};
```

```

        unsigned int maxNumVciBitsSupported;
        unsigned int maxNumActiveVccsAllowed;
        unsigned int maxNumActiveVccsNearEnd;
    } Profile;
case FALSE:
    void;
    } OptionalProfile
} VpInfo<>;
};
struct addVb5VpsRequestAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 14
*****/
This transaction is used to notify the peer OS of the addition of a VP or a VC cross connection associated with a
VB5 interface. The egress direction is out of the Access Network towards Service Node. The ingress direction is
into the Access Network from the Service Node.
*/
struct addVb5ConnectionIndicationCall {
    /* Cell Rate is in cells per second. */
    /* Cell Delay Variation is in microseconds. */
    /* Burst Size is in cells. */
    unsigned int EgressPeakCellRateCLP0Plus1;
    unsigned int EgressPeakCellRateCLP0;
    unsigned int IngressPeakCellRateCLP0Plus1;
    unsigned int IngressPeakCellRateCLP0;
    unsigned int EgressSustainableCellRateCLP0Plus1;
    unsigned int EgressSustainableCellRateCLP0;
    unsigned int IngressSustainableCellRateCLP0Plus1;
    unsigned int IngressSustainableCellRateCLP0;
    unsigned int EgressCDVToleranceCLP0Plus1;
    unsigned int EgressCDVToleranceCLP0;
    unsigned int IngressCDVToleranceCLP0Plus1;
    unsigned int IngressCDVToleranceCLP0;
    unsigned int EgressMaxBurstSizeCLP0Plus1;
    unsigned int EgressMaxBurstSizeCLP0;
    unsigned int IngressMaxBurstSizeCLP0Plus1;
    unsigned int IngressMaxBurstSizeCLP0;
    enum {
        Class0    = 0, /* best effort without specified parameters */
        Class1    = 1, /* parameters as specified for CBR */
        Class2    = 2, /* parameters as specified for VBR */
        Class3    = 3, /* parameters as specified for connection-oriented data */
        Class4    = 4, /* parameters as specified for connectionless data */
    } EgressQosClass;
    enum {
        Class0    = 0, /* best effort without specified parameters */
        Class1    = 1, /* parameters as specified for CBR */
        Class2    = 2, /* parameters as specified for VBR */
        Class3    = 3, /* parameters as specified for connection-oriented data */
        Class4    = 4, /* parameters as specified for connectionless data */
    } IngressQosClass;
    union switch (bool VcCrossConnection) {
        case TRUE:
            struct {
                unsigned int VciValueA; /* VCI on side A */
                unsigned int VciValueB; /* VCI on side B */
            } VciPoints;
        case FALSE:
            void;
    } ConnectionTypeDetails;
};

```

```

unsigned int PhysicalPortA;
unsigned int VpiValueA;
union switch (bool DetailsForPortA) {
    case TRUE:
        struct {
            unsigned int *LogicalServicePort;
            unsigned int *VpciValue;
            /* cannot be specified for both ports */
        } Details;
    case FALSE:
        void;
    } OptionalDetailsForPortA;
unsigned int PhysicalPortB;
unsigned int VpiValueB;
union switch (bool DetailsForPortB) {
    case TRUE:
        struct {
            unsigned int *LogicalUserPort;
            unsigned int *VpciValue;
            /* cannot be specified for both ports */
        } Details;
    case FALSE:
        void;
    } OptionalDetailsForPortB;
} AddVb5ConnectionIndicationCall;
};
struct addVb5ConnectionIndicationAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 15
*****/

This transaction is used to request the peer OS to add a VP or a VC cross connection associated with a VB5
interface. The egress direction is out of the Access Network towards Service Node. The ingress direction is into the
Access Network from the Service Node.
*/
struct addVb5ConnectionRequestCall {
    /* Cell Rate is in cells per second. */
    /* Cell Delay Variation is in microseconds. */
    /* Burst Size is in cells. */
    unsigned int EgressPeakCellRateCLP0Plus1;
    unsigned int EgressPeakCellRateCLP0;
    unsigned int IngressPeakCellRateCLP0Plus1;
    unsigned int IngressPeakCellRateCLP0;
    unsigned int EgressSustainableCellRateCLP0Plus1;
    unsigned int EgressSustainableCellRateCLP0;
    unsigned int IngressSustainableCellRateCLP0Plus1;
    unsigned int IngressSustainableCellRateCLP0;
    unsigned int EgressCDVToleranceCLP0Plus1;
    unsigned int EgressCDVToleranceCLP0;
    unsigned int IngressCDVToleranceCLP0Plus1;
    unsigned int IngressCDVToleranceCLP0;
    unsigned int EgressMaxBurstSizeCLP0Plus1;
    unsigned int EgressMaxBurstSizeCLP0;
    unsigned int IngressMaxBurstSizeCLP0Plus1;
    unsigned int IngressMaxBurstSizeCLP0;
    enum {
        Class0      = 0, /* best effort without specified parameters */
        Class1      = 1, /* parameters as specified for CBR */
        Class2      = 2, /* parameters as specified for VBR */
        Class3      = 3, /* parameters as specified for connection-oriented data */
        Class4      = 4  /* parameters as specified for connectionless data */
    };
};

```

```

    } EgressQosClass;
enum {
    Class0      = 0, /* best effort without specified parameters */
    Class1      = 1, /* parameters as specified for CBR */
    Class2      = 2, /* parameters as specified for VBR */
    Class3      = 3, /* parameters as specified for connection-oriented data */
    Class4      = 4 /* parameters as specified for connectionless data */
} IngressQosClass;
union switch (bool VcCrossConnection) {
    case TRUE:
        struct {
            unsigned int VciValueA; /* VCI on side A */
            unsigned int VciValueB; /* VCI on side B */
        } VciPoints;
    case FALSE:
        void;
} ConnectionTypeDetails;
unsigned int PhysicalPortA;
unsigned int VpiValueA;
union switch (bool DetailsForPortA) {
    case TRUE:
        struct {
            unsigned int *LogicalServicePort;
            unsigned int *VpciValue;
            /* cannot be specified for both ports */
        } Details;
    case FALSE:
        void;
} OptionalDetailsForPortA;
unsigned int PhysicalPortB;
unsigned int VpiValueB;
union switch (bool DetailsForPortB) {
    case TRUE:
        struct {
            unsigned int *LogicalUserPort;
            unsigned int *VpciValue;
            /* cannot be specified for both ports */
        } Details;
    case FALSE:
        void;
} OptionalDetailsForPortB;
} AddVb5ConnectionRequestCall;
};
struct addVb5ConnectionRequestAnswer {
    bool Acknowledge;
};

```

/******

*** Parameter Definitions for Transaction 16**

This transaction is used to request the peer OS to audit a connection which is associated with a VB5 interface.

*/

```

struct auditVb5ConnectionRequestCall {
    unsigned int LogicalServicePortNumber;
    union switch (bool VpCrossConnection) {
        case TRUE:
            struct {
                unsigned int PhysicalPort;
                unsigned int VpiValue;
            } KnownVpEnd;
        case FALSE:
            struct {
                unsigned int PhysicalPort;
            }
    }
};

```

```

        unsigned int VpiValue;
        unsigned int VciValue;
    } KnownVcEnd;
} KnownEnd;
};

struct auditVb5ConnectionRequestAnswer {
    union switch (bool Success) {
        case FALSE:
            enum {
                Unspecified           = 0,
                UnknownPhysicalPort    = 1,
                UnknownVpiValue        = 2,
                UnknownVciValue        = 3
            } FailureReason;
        case TRUE:
            union switch (bool VpCrossConnection) {
                case TRUE:
                    struct {
                        unsigned int PhysicalPort;
                        unsigned int VpiValue;
                    } OtherVpEnd;
                case FALSE:
                    struct {
                        unsigned int PhysicalPort;
                        unsigned int VpiValue;
                        unsigned int VciValue;
                    } OtherVcEnd;
            } OtherEnd;
    } AuditVb5ConnectionRequestAnswer;
};

/*****
* Parameter Definitions for Transaction 17
*****/

This transaction is used to request the peer OS to audit a VPCI which is associated with a VB5 interface.
*/
struct auditVb5VpciRequestCall {
    unsigned int LogicalServicePortNumber;
    union switch (bool KnownVpci) {
        case TRUE:
            struct {
                unsigned int *LogicalUserPortNumber;
                /* present if VPC reaches a user port */
                unsigned int VpciValue;
            } KnownNniVpci;
        case FALSE:
            struct {
                unsigned int PhysicalPort;
                unsigned int VpiValue;
            } KnownRemoteVp;
    } KnownEnd;
};

struct auditVb5VpciRequestAnswer {
    union switch (bool Success) {
        case FALSE:
            enum {
                Unspecified           = 0,
                UnknownLupNumber      = 1,
                UnknownVpci           = 2,
                UnknownPhysicalPort    = 3,
                UnknownVpiValue        = 4
            } FailureReason;
        case TRUE:

```

```

        union switch (bool ReturnedVpci) {
            case TRUE:
                struct {
                    unsigned int *LogicalUserPortNumber;
                    /* present if VPC reaches user port */
                    unsigned int VpciValue
                } ReturnedNniVpci;
            case FALSE:
                struct {
                    unsigned int PhysicalPort;
                    unsigned int VpiValue;
                } ReturnedRemoteVp;
        } ReturnedEnd;
    } AuditVb5VpciRequestAnswer;
};

/*****
* Parameter Definitions for Transaction 18
*****/
This transaction is used to request the peer OS to list the logical user ports associated with a VB5 interface
between an Access Network and a Service Node which the two Operations Systems together control.
*/
struct listLupsRequestCall {
    unsigned int LogicalServicePortNumber;
};
struct listLupsRequestAnswer {
    union switch (bool Success) {
        case FALSE:
            enum {
                Unspecified          = 0,
                UnknownLspNumber     = 1
            } FailureReason;
        case TRUE:
            unsigned int LogicalUserPortNumber<>;
    } ListLupsRequestAnswer;
};

/*****
* Parameter Definitions for Transaction 19
*****/
This transaction is used to request the peer OS to list the details of the protocols of a VB5 interface between an
Access Network and a Service Node which the two Operations Systems together control.
*/
struct listVb5ProtocolDetailsRequestCall {
    unsigned int LogicalServicePortNumber;
};
struct listVb5ProtocolDetailsRequestAnswer {
    union switch (bool Success) {
        case FALSE:
            enum {
                Unspecified          = 0,
                UnknownLspNumber     = 1
            } FailureReason;
        case TRUE:
            struct {
                enum {
                    RTMC          = 1,
                    BBCC          = 2
                } ProtocolType;
                unsigned int VpciValue;
                unsigned int VciValue;
                struct {
                    /* Cell Rate is in cells per second. */

```

```

        /* Cell Delay Variation is in microseconds. */
        /* Burst Size is in cells. */
        unsigned int EgressPeakCellRateCLP0Plus1;
        unsigned int EgressPeakCellRateCLP0;
        unsigned int IngressPeakCellRateCLP0Plus1;
        unsigned int IngressPeakCellRateCLP0;
        unsigned int EgressSustainableCellRateCLP0Plus1;
        unsigned int EgressSustainableCellRateCLP0;
        unsigned int IngressSustainableCellRateCLP0Plus1;
        unsigned int IngressSustainableCellRateCLP0;
        unsigned int EgressCDVToleranceCLP0Plus1;
        unsigned int EgressCDVToleranceCLP0;
        unsigned int IngressCDVToleranceCLP0Plus1;
        unsigned int IngressCDVToleranceCLP0;
        unsigned int EgressMaxBurstSizeCLP0Plus1;
        unsigned int EgressMaxBurstSizeCLP0;
        unsigned int IngressMaxBurstSizeCLP0Plus1;
        unsigned int IngressMaxBurstSizeCLP0;
    } QosParameters;
    struct {
        bool BufferRelease; /* on connection release */
        unsigned int MaxCc;
            /* maximum value in PDUs of the state variable VT(CC) */
        unsigned int MaxInformationFieldLength; /* in octets */
        unsigned int MaxLengthScopUuField; /* in octets */
        unsigned int MaxPd;
            /* maximum value in PDUs of the state variable VT(PD) */
        unsigned int MaxScopCreditToPeer;
            /* absolute value in PDUs of the receive window */
        unsigned int MaxStat;
            /* maximum number of list elements in a STAT PDU */
        unsigned int SscopTimerCc;
            /* time interval between transmissions in milliseconds */
        unsigned int SscopTimerIdle; /* idle phase in milliseconds */
        unsigned int SscopTimerKeepAlive;
            /* keep alive phase in milliseconds */
        unsigned int SscopTimerNoResponse;
            /* no response time in milliseconds */
        unsigned int SscopTimerPoll;
            /* active POLL phase in milliseconds */
    } SaalParameters;
    } ProtocolDetails<>;
} ListProtocolDetailsRequestAnswer;
};

/*****
* Parameter Definitions for Transaction 20
*****/

This transaction is used to request the peer OS to list the identities of the VB5 interfaces between Access
Network(s) and the Service Node(s) which the two Operations Systems together control.
*/
struct listVb5InterfacesRequestCall {
    void;
};
struct listVb5InterfacesRequestAnswer {
    unsigned int LogicalServicePortNumber<>;
};

```



```

/*****

```

``` * Parameter Definitions for Transaction 21 ```

```

****

```

This transaction is used to request the peer OS to list the VCs associated with a VB5 interface.

```

*/

```

```

struct listVb5VcsRequestCall {
    unsigned int LogicalServicePortNumber;
    unsigned int *LogicalUserPortNumber;
    /* present if response constrained to a particular LUP */
    unsigned int *VpciValue;
    /* present if response constrained to a particular VPCI */
};

```

```

struct listVb5VcsRequestAnswer {
    union switch (bool Success) {
        case FALSE:
            enum {
                Unspecified                = 0,
                UnknownLspNumber           = 1,
                UnknownLupNumber           = 2,
                UnknownVpciValue           = 3,
                UnknownLupVpciCombination   = 4
            } FailureReason;
        case TRUE:
            struct {
                unsigned int *LogicalUserPort;
                /* present if VCC reached LUP */
                unsigned int VpciValue;
                unsigned int VciValue<>;
            } VpciVcs<>;
    } ListVb5VcsRequestAnswer;
};

```

```

/*****

```

``` * Parameter Definitions for Transaction 22 ```

```

****

```

This transaction is used to request the peer OS to list the VPs associated with a VB5 interface.

```

*/

```

```

struct listVb5VpsRequestCall {
    unsigned int LogicalServicePortNumber;
    union switch (bool AllVps)
        case TRUE:
            void;
        case FALSE:
            unsigned int *LogicalUserPortNumber;
            /* absent for selection of VPCs terminating in the AN */
    } Filter;
    unsigned int *LogicalUserPortNumber;
};

```

```

struct listVb5VpsRequestAnswer {
    union switch (bool Success) {
        case FALSE:
            enum {
                Unspecified                = 0,
                UnknownLspNumber           = 1,
                UnknownLupNumber           = 2
            } FailureReason;
        case TRUE:
            struct
                unsigned int *LogicalUserPort;
                /* present if VPC reaches LUP */
            struct {
                unsigned int PhysicalPort;
                unsigned int VpiValue;
            }
    }
};

```

```

        unsigned int VpciValue;
        union switch (bool ProfileSupported) {
            case TRUE:
                struct {
                    unsigned int maxNumVciBitsNearEnd;
                    unsigned int maxNumVciBitsSupported;
                    unsigned int maxNumActiveVccsAllowed;
                    unsigned int maxNumActiveVccsNearEnd;
                } Profile;
            case FALSE:
                void;
        } OptionalProfile
    } VpInfo;
} SubportVps<>;
} ListVb5VpsRequestAnswer;
};

/*****
* Parameter Definitions for Transaction 23
*****/
This transaction is used by the OS of the SN to request the OS of the AN to remove a loop from a connection.
*/
struct removeAnLoopRequestCall {
    unsigned int LogicalServicePort;
    union switch (bool VpcReachesUserPort) {
        case TRUE:
            unsigned int LogicalUserPort;
        case FALSE:
            void;
    } OptionalUserPort;
    unsigned int VpciValue;
    union switch (bool VccLoopback) {
        case TRUE:
            unsigned int VciValue;
        case FALSE:
            void;
    } OptionalVciValue;
};

struct removeAnLoopRequestAnswer {
    union switch (bool Failure) {
        case TRUE:
            enum {
                Unspecified                = 0,
                UnknownLogicalServicePort   = 1,
                UnknownLogicalUserPort      = 2,
                UnknownVpciValue            = 3,
                UnknownVciValue             = 4,
                NoLoopPresent                = 5
            } FailureReason;
        case FALSE:
            void;
    } OptionalFailureReason;
};

/*****
* Parameter Definitions for Transaction 24
*****/
This transaction is used to notify the peer OS of the removal of logical user ports from a VB5 interface.
*/
struct removeLupsIndicationCall {
    unsigned int LogicalServicePortNumber;
    unsigned int LogicalUserPortNumber<>;
};

```

```

struct removeLupsIndicationAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 25
*****/
This transaction is used to request the peer OS to remove logical user ports from a VB5 interface.
*/
struct removeLupsRequestCall {
    unsigned int LogicalServicePortNumber;
    unsigned int LogicalUserPortNumber<>;
};
struct removeLupsRequestAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 26
*****/
This transaction is used to notify the peer OS of the removal of the protocol VP from a VB5 interface.
*/
struct removeVb5ProtocolVpIndicationCall {
    unsigned int LogicalServicePortNumber;
    unsigned int VpciValue;
};
struct removeVb5ProtocolVpIndicationAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 27
*****/
This transaction is used to request the peer OS to remove the protocol VP from a VB5 interface.
*/
struct removeVb5ProtocolVpRequestCall {
    unsigned int LogicalServicePortNumber;
    unsigned int VpciValue;
};
struct removeVb5ProtocolVpRequestAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 28
*****/
This transaction is used to notify the peer OS of the removal of a VB5 interface.
*/
struct removeVb5InterfaceIndicationCall {
    unsigned int LogicalServicePortNumber;
};
struct removeVb5InterfaceIndicationAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 29
*****/
This transaction is used to request the peer OS to remove a VB5 interface.
*/
struct removeVb5InterfaceRequestCall {
    unsigned int LogicalServicePortNumber;
};

```

```

struct removeVb5InterfaceRequestAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 30
*****/
This transaction is used to notify the peer OS of the removal of VCs from a VP which is associated with a VB5
interface.
*/
struct removeVb5VcsIndicationCall {
    unsigned int LogicalServicePortNumber;
    unsigned int *LogicalServiceSubport;
    /* present if VPC reaches a user port */
    unsigned int VpciValue;
    unsigned int VciValue<>;
};
struct removeVb5VcsIndicationAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 31
*****/
This transaction is used to request the peer OS to remove VCs from a VP which is associated with a VB5
interface.
*/
struct removeVb5VcsRequestCall {
    unsigned int LogicalServicePortNumber;
    unsigned int *LogicalServiceSubport;
    /* present if VPC reaches a user port */
    unsigned int VpciValue;
    unsigned int VciValue<>;
};
struct removeVb5VcsRequestAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 32
*****/
This transaction is used to notify the peer OS of the removal of a protocol from a VB5 interface.
*/
struct removeVb5ProtocolIndicationCall {
    unsigned int LogicalServicePortNumber;
    enum {
        RTMC      = 1,
        BBCC      = 2
    } ProtocolType;
};
struct removeVb5ProtocolIndicationAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 33
*****/
This transaction is used to request the peer OS to remove a protocol from a VB5 interface.
*/
struct removeVb5ProtocolRequestCall {
    unsigned int LogicalServicePortNumber;
    enum {
        RTMC      = 1,

```

```

        BBCC      = 2
    } ProtocolType;
};
struct removeVb5ProtocolRequestAnswer {
    bool Acknowledge;

    /**
     * Parameter Definitions for Transaction 34
     *
     * This transaction is used to notify the peer OS of the removal of VPs that are associated with a VB5 interface.
     */
    struct removeVb5VpsIndicationCall {
        unsigned int LogicalServicePortNumber;
        unsigned int *LogicalServiceSubport;
        /* present if VPC reaches a user port */
        struct {
            unsigned int PhysicalPort;
            unsigned int VpiValue;
            unsigned int VpciValue;
        } VpInfo<>;
    };
    struct removeVb5VpsIndicationAnswer {
        bool Acknowledge;
    };

    /**
     * Parameter Definitions for Transaction 35
     *
     * This transaction is used to request the peer OS to remove VPs that are associated with a VB5 interface.
     */
    struct removeVb5VpsRequestCall {
        unsigned int LogicalServicePortNumber;
        unsigned int *LogicalServiceSubport;
        /* present if VPC reaches a user port */
        struct {
            unsigned int PhysicalPort;
            unsigned int VpiValue;
            unsigned int VpciValue;
        } VpInfo<>;
    };
    struct removeVb5VpsRequestAnswer {
        bool Acknowledge;
    };

    /**
     * Parameter Definitions for Transaction 36
     *
     * This transaction is used to notify the peer OS of the removal of a VP or a VC cross connection associated with a
     * VB5 interface. The egress direction is out of the Access Network towards Service Node. The ingress direction is
     * into the Access Network from the Service Node.
     */
    struct removeVb5ConnectionIndicationCall {
        union switch (bool VcCrossConnection) {
            case TRUE:
                struct {
                    unsigned int VciValueA; /* VCI on side A */
                    unsigned int VciValueB; /* VCI on side B */
                } VciPoints;
            case FALSE:
                void;
        } ConnectionTypeDetails;
        unsigned int PhysicalPortA;
        unsigned int VpiValueA;
    };

```

```

union switch (bool DetailsForPortA) {
    case TRUE:
        struct {
            unsigned int *LogicalServicePort;
            unsigned int *VpciValue;
            /* cannot be specified for both ports */
        } Details;
    case FALSE:
        void;
    } OptionalDetailsForPortA;
unsigned int PhysicalPortB;
unsigned int VpiValueB;
union switch (bool DetailsForPortB) {
    case TRUE:
        struct {
            unsigned int *LogicalUserPort;
            unsigned int *VpciValue;
            /* cannot be specified for both ports */
        } Details;
    case FALSE:
        void;
    } OptionalDetailsForPortB;
} RemoveVb5ConnectionIndicationCall;
};
struct removeVb5ConnectionIndicationAnswer {
    bool Acknowledge;
};

```

/******

*** Parameter Definitions for Transaction 37**

This transaction is used to request the peer OS to remove a VP or a VC cross connection associated with a VB5 interface. The egress direction is out of the Access Network towards Service Node. The ingress direction is into the Access Network from the Service Node.

*/

```

struct removeVb5ConnectionRequestCall {
    union switch (bool VcCrossConnection) {
        case TRUE:
            struct {
                unsigned int VciValueA; /* VCI on side A */
                unsigned int VciValueB; /* VCI on side B */
            } VciPoints;
        case FALSE:
            void;
        } ConnectionTypeDetails;
unsigned int PhysicalPortA;
unsigned int VpiValueA;
union switch (bool DetailsForPortA) {
    case TRUE:
        struct {
            unsigned int *LogicalServicePort;
            unsigned int *VpciValue;
            /* cannot be specified for both ports */
        } Details;
    case FALSE:
        void;
    } OptionalDetailsForPortA;
unsigned int PhysicalPortB;
unsigned int VpiValueB;
union switch (bool DetailsForPortB) {
    case TRUE:
        struct {
            unsigned int *LogicalUserPort;

```

```

        unsigned int *VpciValue;
        /* cannot be specified for both ports */
    } Details;
case FALSE:
    void;
    } OptionalDetailsForPortB;
} RemoveVb5ConnectionRequestCall;
};
struct removeVb5ConnectionRequestAnswer {
    bool Acknowledge;
};

/*****
* Parameter Definitions for Transaction 38
*****/
This transaction is used by the OS of an AN to request the label that an SN uses for the AN.
*/
struct anServiceLabelInquiryCall {
    void;
};
struct anServiceLabelInquiryAnswer {
    union switch (bool AnLabelKnown) {
        case TRUE:
            unsigned int AnLabel;
        case FALSE:
            void;
    } AnServiceLabelInquiryAnswer;
};

/*****
* Parameter Definitions for Transaction 39
*****/
This transaction is used by the OS of an SN to request the label that an AN uses for the SN and the local reference
that the AN uses for the VB5 interface.
*/
struct snAccessLabelsInquiryCall {
    void;
};
struct snAccessLabelsInquiryAnswer {
    union switch (bool SnAndIntLabelsKnown) {
        case TRUE:
            unsigned int SnLabel;
            unsigned int InterfaceLabel;
        case FALSE:
            void;
    } snAccessLabelsInquiryAnswer;
};

/*****
* Parameter Definitions for Transaction 40
*****/
This transaction is used to notify the peer OS of the change of status of a Resource.
*/
struct resourceStatusIndicationCall {
    unsigned int LogicalServicePort;
    union switch (bool SpecifiedLogicalUserPort) {
        case TRUE:
            unsigned int LogicalServicePort;
        case FALSE:
            void;
    } OptionalLogicalServicePort;
    unsigned int VpciValue<>;
    enum {

```

```

        FullyOperational           = 0,
        AdministratelyBlockedTestCallsAllowed = 1,
        AdministratelyBlockedNoCellFlow      = 2,
        Fault                             = 3
    } Status;
};
struct resourceStatusIndicationAnswer {
    bool Acknowledge;
};

```

APÉNDICE I

Bibliografía

- [1] UIT-T G.773 (1993), *Series de protocolos de interfaces Q para la gestión de sistemas de transmisión.*
- [2] UIT-T G.774 (2001), *Modelo de información de gestión de la jerarquía digital síncrona desde el punto de vista de los elementos de red.*
- [3] UIT-T G.803 (2000), *Arquitecturas de redes de transporte basadas en la jerarquía digital síncrona.*
- [4] UIT-T I.211 (1993), *Aspectos de servicio de la red digital de servicios integrados de banda ancha.*
- [5] UIT-T I.311 (1996), *Aspectos generales de la red digital de servicios integrados de banda ancha (RDSI-BA).*
- [6] UIT-T I.327 (1993), *Arquitectura funcional de la red digital de servicios integrados de banda ancha.*
- [7] UIT-T I.356 (2000), *Calidad de funcionamiento en la transferencia de células en la capa de modo de transferencia asíncrono de la red digital de servicios integrados de banda ancha.*
- [8] UIT-T I.371 (2000), *Control de tráfico y control de congestión en RDSI-BA.*
- [9] UIT-T I.413 (1993), *Interfaz usuario-red de la red digital de servicios integrados de banda ancha.*
- [10] UIT-T de la serie I.432.x, *Interfaz usuario-red de la red digital de servicios integrados de banda ancha (RDSI-BA) – Especificación de la capa física.*
- [11] UIT-T I.580 (1995), *Disposiciones generales para el interfuncionamiento entre la red digital de servicios integrados de banda ancha y la red digital de servicios integrados basada en la velocidad de 64 kbit/s.*
- [12] UIT-T I.610 (1999), *Principios y funciones de operaciones y mantenimiento de la red digital de servicios integrados de banda ancha.*
- [13] UIT-T I.732 (2000), *Características funcionales del equipo del modo transferencia asíncrono.*
- [14] UIT-T M.3200 (1997), *Servicios de gestión de red de gestión de las telecomunicaciones y sectores gestionados de las telecomunicaciones: Panorama general.*
- [15] UIT-T M.3207.1 (1996), *Servicio de gestión de la red de gestión de las telecomunicaciones: Aspectos de mantenimiento de la gestión de la red digital de servicios integrados de banda ancha.*
- [16] UIT-T M.3400 (2000), *Funciones de gestión de la red de gestión de las telecomunicaciones.*

- [17] UIT-T M.3610 (1996), *Principios de aplicación del concepto de red de gestión de las telecomunicaciones a la gestión de la red digital de servicios integrados de banda ancha.*
- [18] UIT-T Q.821 (2000), *Descripción de las etapas 2 y 3 de la interfaz Q3 – Vigilancia de alarmas.*
- [19] UIT-T Q.822 (1994), *Descripción de la etapa 1, de la etapa 2 y de la etapa 3 para la interfaz Q3 – Gestión de la calidad de funcionamiento.*
- [20] UIT-T X.208 (1988), *Especificación de la notación de sintaxis abstracta uno (ASN.1).*
- [21] UIT-T X.701 (1997) | ISO/CEI 10040:1998, *Tecnología de la información – Interconexión de sistemas abiertos – Visión general de la gestión de sistemas.*
- [22] UIT-T X.722 (1992) | ISO/CEI 10165-4:1992, *Tecnología de la información – Interconexión de sistemas abiertos – Estructura de la información de gestión: Directrices para la definición de objetos gestionados.*
- [23] UIT-T X.733 (1992) | ISO/CEI 10164-4:1992, *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función señaladora de alarmas.*
- [24] UIT-T X.734 (1992) | ISO/CEI 10164-5:1993, *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de gestión de informes de eventos.*
- [25] UIT-T X.735 (1992) | ISO/CEI 10164-6:1993, *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función control de ficheros registro cronológico.*
- [26] UIT-T X.737 (1995) | ISO/CEI 10164-14:1996, *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Categorías de pruebas de confianza y de diagnóstico.*
- [27] UIT-T X.738 (1993) | ISO/CEI 10164-13:1995, *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de sumario.*
- [28] UIT-T X.739 (1993) | ISO/CEI 10164-11:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Objetos métricos y atributos.*
- [29] UIT-T X.745 (1993) | ISO/CEI 10164-12:1994, *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de gestión de prueba.*
- [30] UIT-T X.746 (1995) | ISO/CEI 10164-15:1995, *Tecnología de la información – Interconexión de sistemas abiertos – Gestión de sistemas: Función de planificación.*
- [31] Especificación del Foro ATM, *M4 Interface Requirements and Logical MIB; ATM Network Element View, Versión 1.0.*
- [32] Especificación del Foro ATM, *CMIP Specification for the M4 Interface, Versión 1.0.*
- [33] Especificación del Foro ATM, *ATM User-Network Interface Specification, Versión 3.0.*
- [34] Especificación del Foro ATM, *ATM User-Network Interface Specification, Versión 3.1.*
- [35] Especificación del Foro ATM, *ATM User-Network Interface Specification, Versión 4.0.*

APÉNDICE II

Sumario de transacciones

Categoría	Nombre	Tipo	Origen
Creación de VB5	anServiceLabelInquiry	Acción	AN
	snAccessLabelsInquiry	Acción	SN
	addVb5InterfaceRequest	Acción	ambos
	addVb5InterfaceIndication	Notificación	ambos
Supresión de VB5	removeVb5InterfaceRequest	Acción	ambos
	removeVb5InterfaceIndication	Notificación	ambos
Modificación – VP protocolo VB5	addVb5ProtocolVpRequest	Acción	ambos
	addVb5ProtocolIndication	Notificación	ambos
	removeVb5ProtocolVpRequest	Acción	ambos
	removeVb5ProtocolVpIndication	Notificación	ambos
Modificación – Protocolos VB5	addVb5ProtocolVpRequest	Acción	ambos
	addVb5ProtocolVpIndication	Notificación	ambos
	removeVb5ProtocolRequest	Acción	ambos
	removeVb5ProtocolIndication	Notificación	ambos
Modificación – Puertos de usuario lógicos	addLupsRequest	Acción	ambos
	addLupsIndication	Notificación	ambos
	removeLupsRequest	Acción	ambos
	removeLupsIndication	Notificación	ambos
Modificación – VP	addVb5VpsRequest	Acción	ambos
	addVb5VpsIndication	Notificación	ambos
	removeVb5VpsRequest	Acción	ambos
	removeVb5VpsIndication	Notificación	ambos
Modificación – VC	addVb5VcsRequest	Acción	ambos
	addVb5VcsIndication	Notificación	ambos
	removeVb5VcsRequest	Acción	ambos
	removeVb5VcsIndication	Notificación	ambos
Modificación – Conexiones	addVb5ConnectionRequest	Acción	ambos
	addVb5ConnectionIndication	Notificación	ambos
	removeVb5ConnectionRequest	Acción	ambos
	removeVb5ConnectionIndication	Notificación	ambos
Auditoría – Peticiones de listas	listVb5InterfacesRequest	Acción	ambos
	listVb5ProtocolDetailsRequest	Acción	ambos
	listLupsRequest	Acción	ambos
	listVb5VpsRequest	Acción	ambos
	listVb5VcsRequest	Acción	ambos
Auditoría – Peticiones de auditoría	auditVb5ConnectionRequest	Acción	ambos
	auditVb5VpciRequest	Acción	ambos
Informes de averías	resourceStatusIndication	Notificación	ambos
Localización de averías	addAnLoopRequest	Acción	SN
	removeAnLoopRequest	Acción	SN

SERIES DE RECOMENDACIONES DEL UIT-T

Serie A	Organización del trabajo del UIT-T
Serie B	Medios de expresión: definiciones, símbolos, clasificación
Serie C	Estadísticas generales de telecomunicaciones
Serie D	Principios generales de tarificación
Serie E	Explotación general de la red, servicio telefónico, explotación del servicio y factores humanos
Serie F	Servicios de telecomunicación no telefónicos
Serie G	Sistemas y medios de transmisión, sistemas y redes digitales
Serie H	Sistemas audiovisuales y multimedios
Serie I	Red digital de servicios integrados
Serie J	Redes de cable y transmisión de programas radiofónicos y televisivos, y de otras señales multimedios
Serie K	Protección contra las interferencias
Serie L	Construcción, instalación y protección de los cables y otros elementos de planta exterior
Serie M	RGT y mantenimiento de redes: sistemas de transmisión, circuitos telefónicos, telegrafía, facsímil y circuitos arrendados internacionales
Serie N	Mantenimiento: circuitos internacionales para transmisiones radiofónicas y de televisión
Serie O	Especificaciones de los aparatos de medida
Serie P	Calidad de transmisión telefónica, instalaciones telefónicas y redes locales
Serie Q	Conmutación y señalización
Serie R	Transmisión telegráfica
Serie S	Equipos terminales para servicios de telegrafía
Serie T	Terminales para servicios de telemática
Serie U	Conmutación telegráfica
Serie V	Comunicación de datos por la red telefónica
Serie X	Redes de datos y comunicación entre sistemas abiertos
Serie Y	Infraestructura mundial de la información y aspectos del protocolo Internet
Serie Z	Lenguajes y aspectos generales de soporte lógico para sistemas de telecomunicación