ETSI EN 301 754 V1.1.1 (2001-09)

European Standard (Telecommunications series)

Telecommunications Management Network (TMN); Management interfaces associated with the VB5.2 reference point

Reference

DEN/TMN-00003

Keywords

TMN, Q3 interface, ATM, configuration, management, performance, V5 interface, OAM

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This European Standard (Telecommunications series) has been produced by ETSI Technical Committee Telecommunications Management Network (TMN).

National transposition dates	
Date of adoption of this EN:	20 July 2001
Date of latest announcement of this EN (doa):	31 October 2001
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	30 April 2002
Date of withdrawal of any conflicting National Standard (dow):	30 April 2002

1 Scope

The present document specifies the management interfaces (Q3 interfaces) associated with the VB5.2 reference point [6] and EN 301 217-1 (see bibliography) for the support of configuration, fault & performance management functions. Fault and performance management together include both passive monitoring of reports and active fault isolation.

The Q3 interface is the TMN interface between network elements or Q-adapters which interface to OSs without mediation and between OSs and mediation devices.

Existing protocols are used where possible, and the focus of the work is on defining the object model. The definition of the functionality of TMN Operations Systems is outside the scope of the present document.

ITU-T Recommendation Q.2931 [9] is supported at the UNI, and the ATM Forum UNI is supported for compatibility with the established base of ATM equipment.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] ETSI EN 301 271: "Telecommunications Management Network (TMN); Management interfaces associated with the VB5.1 reference point".

[2] ITU-T Recommendation G.773: "Protocol suites for Q-interfaces for management of transmission systems".

[3] ITU-T Recommendation G.784: "Synchronous digital hierarchy (SDH) management".

[4] ITU-T Recommendation G.902: "Framework Recommendation on functional access networks (AN) Architecture and functions, access types, management and service node aspects".

[5] ITU-T Recommendation G.967.1 (1998): "V-interfaces at the Service Node (SN): VB5.1 reference point specification".

[6] ITU-T Recommendation G.967.2 (1998): "V-interfaces at the Service Node (SN): VB5.2 reference point specification".

[7] ITU-T Recommendation I.751: "Asynchronous transfer mode management of the network element view".

[8] ITU-T Recommendation M.3100 (1995): "Generic network information model".

[9] ITU-T Recommendation Q.2931: "Broadband Integrated Services Digital Network (B-ISDN) - Digital Subscriber Signalling System No. 2 (DSS 2) - User-Network Interface (UNI) - Layer 3 specification for basic call/connection control".

[10] ITU-T Recommendation Q.811: "Lower layer protocol profiles for the Q3 and X interfaces".

[11] ITU-T Recommendation Q.812: "Upper layer protocol profiles for the Q3 and X interfaces".

[12] Void.

[13] ITU-T Recommendation Q.824.6: "Stage 2 and stage 3 description for the Q3 interface - Customer administration: Broadband switch management".

[14] ITU-T Recommendation Q.832.1 (1998): "VB5.1 Management".

[15] ITU-T Recommendation Q.832.2 (1999): "VB5.2 Management".

[16] ITU-T Recommendation X.721 | ISO/IEC 10165-2 (1992): "Information technology - Open Systems Interconnection - Structure of management information: definition of management information".

3 Definitions, abbreviations, and conventions

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

VB5 Resources: management of user port functions and service port functions providing User Network Interface (UNI) and Service Node Interface (SNI) functionality

NOTE: They are respectively considered in EN 301 754 based on the framework defined in ITU-T Recommendation G.902. Transmission specific resources lie outside its scope. VB5 Resources are referred to in the present document as resources

In addition, the present document uses terms defined in ITU-T Recommendations:

ITU-T Recommendation G.902: Access network (AN), User port functions, Service node (SN), Service node interface (SNI), Service port functions.

ITU-T Recommendation G.967.1: Logical service port (LSP), Logical user port (LUP), Physical service port (PSP), Physical user port (PUP), Real-time management coordination (RTMC), Virtual user port (VUP).

ITU-T Recommendation G.967.2: Broadband bearer connection control (B-BCC).

ITU-T Recommendation I.751: Message communication function (MCF).

3.2 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AN	Access Network
ASN.1	Abstract Syntax Notation One
ATM	Asynchronous Transfer Mode
B-BCC	Broadband Bearer Connection Control
GDMO	Guidelines for the Definition Of Managed Objects
LSP	Logical Service Port
LUP	Logical User Port
MIB	Management Information Base
MOC	Managed Object Class
OAM	Operations, Administration and Maintenance
OS	Operations System
RTMC	Real-Time Management Coordination
SDH	Synchronous Digital Hierarchy
SN	Service Node
SNI	Service Node Interface
TMN	Telecommunications Management Network
TTP	Trail Termination Point
UNI	User-Network Interface
VC	Virtual Channel
VP	Virtual Path
VPC	Virtual Path Connection
VPCI	Virtual Path Connection Identifier

3.3 Conventions

NOTE: While this specification on VB5.2 management makes use of ITU-T Recommendation X.722/Amd.1 (1995) on the SET-BY-CREATE property, the reader should be aware that ITU-T Recommendation X.722/Amd.1 (1995) has not been applied in the VB5.1 management specification.

Objects and their characteristics and associated ASN.1 defined here are given names with capitals used to indicate the start of the next word and acronyms are treated as if they were words.

Throughout the present document, all new attributes are named according to the following guidelines:

- The name of an attribute ends in the string "Ptr" if and only if the attribute value is intended to identify a single object.
- The name of an attribute ends in the string "PtrList" if and only if the attribute value is intended to identify one or more objects.
- The name of an attribute is composed of the name of an object class followed by the string "Ptr" if and only if the attribute value is intended to identify a specific object class.
- If an attribute is intended to identify different object classes, a descriptive name is given to that attribute and a description is provided in the attribute behaviour.
- The name of an attribute ends in the string "Id" if and only if the attribute value is intended to identify the name of an object, in which case this attribute should be the first one listed, should use ASN.1 NameType and should not be used to convey other information.
- The name of an attribute is composed of the name of an object class followed by the string "Id" if and only if the attribute value is intended to identify the name of the object class holding that attribute.

4 General Overview

The following information model diagrams have been drawn for the purpose of clarifying the relations between the different object classes of the model.

1) Entity-relationship models showing the relations of the different managed objects.

2) Inheritance Hierarchy showing how managed objects are derived from each other (i.e. the different paths of inherited characteristics of the different managed objects).

These diagrams are only for clarification. The formal specification in terms of GDMO templates and ASN.1 type definitions are the relevant information for implementations.

4.1 Entity-relationship models

The following conventions (see figure 1) are used in the diagrams:

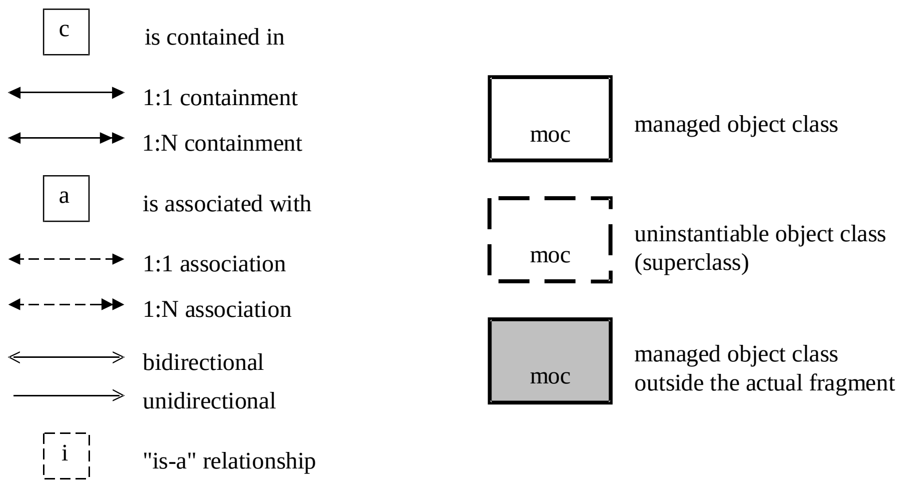
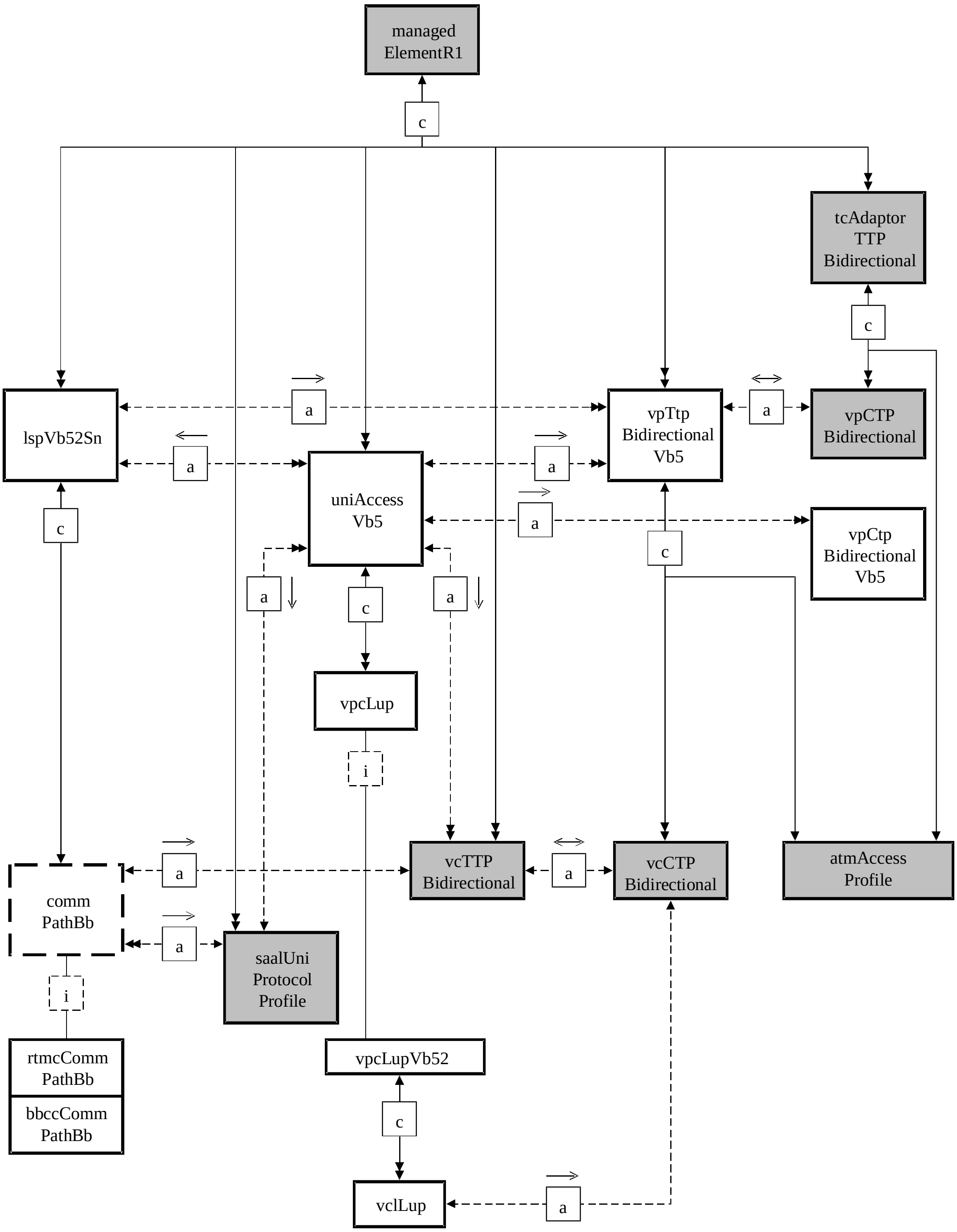


Figure 1: Conventions used in diagrams for Entity-relationship models

Where the directionality of containment is not clear, it can be identified by implications since the root class is unique.

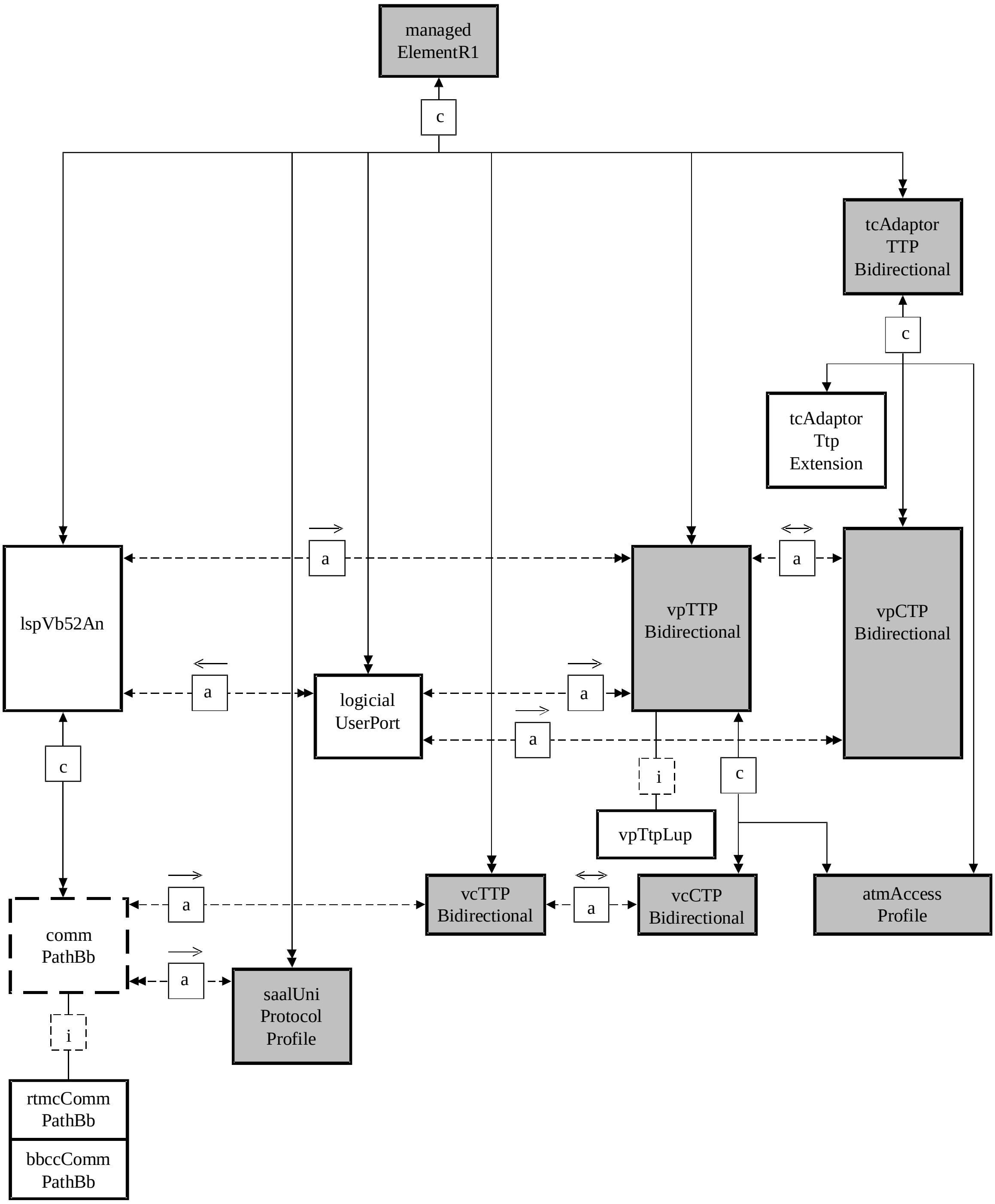
4.1.1 Entity relationship diagram for the service node



NOTE: Not all object classes are shown in this diagram as some object classes are reused unchanged from ITU-T Recommendation I.751 [7].

Figure 2: Entity-relationship diagram - Service node

4.1.2 Entity relationship diagram for the access network



NOTE: Not all object classes are shown in this diagram as some object classes are reused unchanged from ITU-T Recommendation I.751 [7].

Figure 3: Entity-relationship diagram - Access network

4.2 Inheritance hierarchy

Figure 4 traces the inheritance relationships from the highest level object (ITU-T Recommendation X.721 [16], "top") to the managed objects which are defined in the present document.

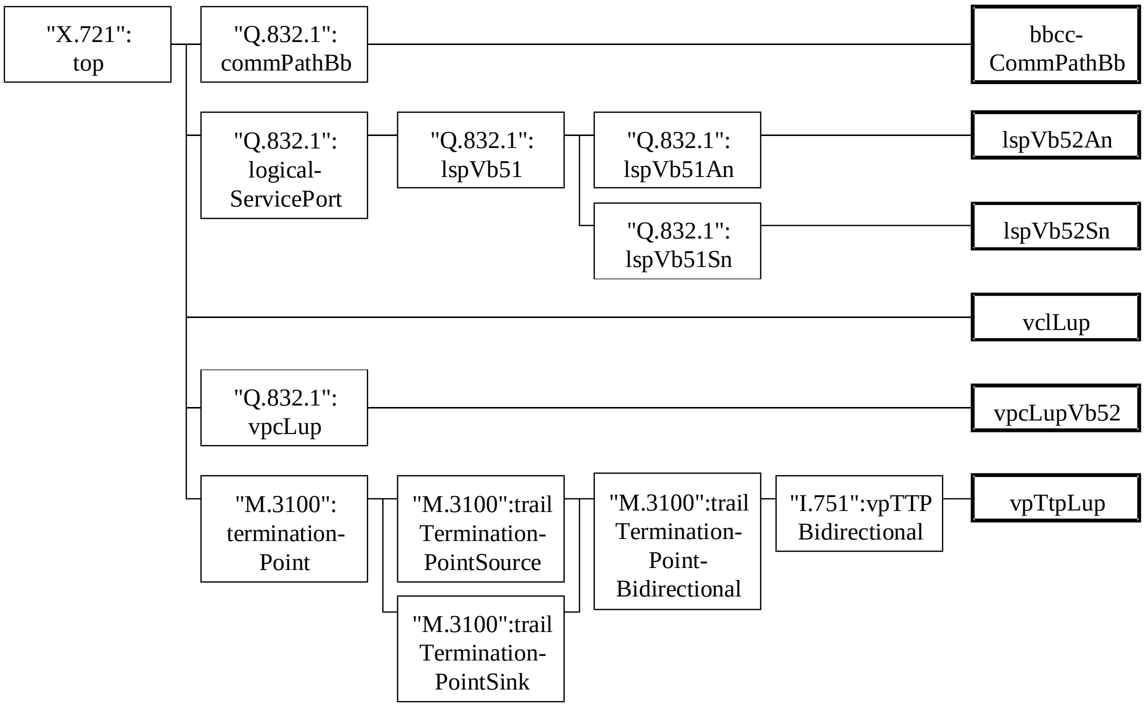


Figure 4: Inheritance hierarchy

5 Formal Definitions

This clause gives the formal definitions of the managed object classes, name bindings, general packages, behaviours, attributes, actions and notifications.

Formal definitions are shown in annex F.

6 Type definitions

Type definitions are shown in annex F.7.

7 Protocol stacks

- The protocol stacks specified in ITU-T Recommendations Q.811 [10], Q.812 [11], G.773 [2] and the SDH digital cross-connect part of ITU-T Recommendation G.784 [3] can be used as part of the protocol stack for the present document.

Annex A (normative): Management requirements

A.1 General management requirements

A.1.1 General configuration management requirements

a) There is a requirement to assign a VB5 interface identifier, also known as a logical service port identifier, to a VB5 interface.

b) There is a requirement to assign VPCIs to VPCs on a VB5 interface when these VPCs are terminated in the access network.

A.2 Real-time management coordination requirements

A.2.1 Configuration management requirements

A.2.1.1 General configuration management requirement

The general configuration requirements include the general real-time management coordination functions between the access network and the service node.

A.2.1.2 Common configuration management requirement for AN and SN

A.2.1.2.1 Shutting-down of VPs

The model should support the MEE primitives associated with the shutting down of VPs.

A.2.1.2.2 VB5 interface ID checking

The management interfaces must support the verification of logical VB5 interface IDs so that the connection of VB5 interfaces can be checked by the operations systems.

A.2.1.2.3 Handling of VB5 primitives

The operations system must be able to handle the MEE primitives in AN and the SN.

A.2.1.2.4 Coordination of VP and VC resources

There is a requirement for the service node to have knowledge of the state of VP and VC resources used to provide service to the customer.

A.2.1.2.5 Non B-ISDN accesses

There is a requirement to take account of VCs terminated in the access network for non-B-ISDN accesses represented by virtual user ports (whose nature is not explicit) and to allow cross connections for these.

A.2.1.3 Configuration management requirement for AN

A.2.1.4 Configuration management requirement for SN

A.2.1.4.1 Assignment of indirect accesses

There is a requirement to assign indirect UNI accesses in the service node to VB5 interfaces at the service node.

A.2.1.4.2 Coordination of indirect accesses with logical user ports

There is requirement to relate indirect UNI accesses in the service node to logical user ports in the access network.

A.2.1.4.3 Consistency of configuration

There is a requirement to check the consistency of the configuration VPCIs between the access network and the service node.

A.2.2 Fault management requirements

A.2.2.1 Alarm surveillance requirements

A.2.2.1.1 General alarm surveillance requirements

A.2.2.1.1.1 Coordination of operational states

Where changes of the operational state of ATM entities are communicated between the access network and the service node using ATM OAM cells, it must be possible to inform the operations systems about these communicated changes since higher management functions may be affected. This is dealt with in ITU-T Recommendation I.751 [7].

A.2.2.2 Test and fault localization requirements

A.2.2.2.1 General test and fault localization requirements

A.2.2.2.1.1 Test traffic

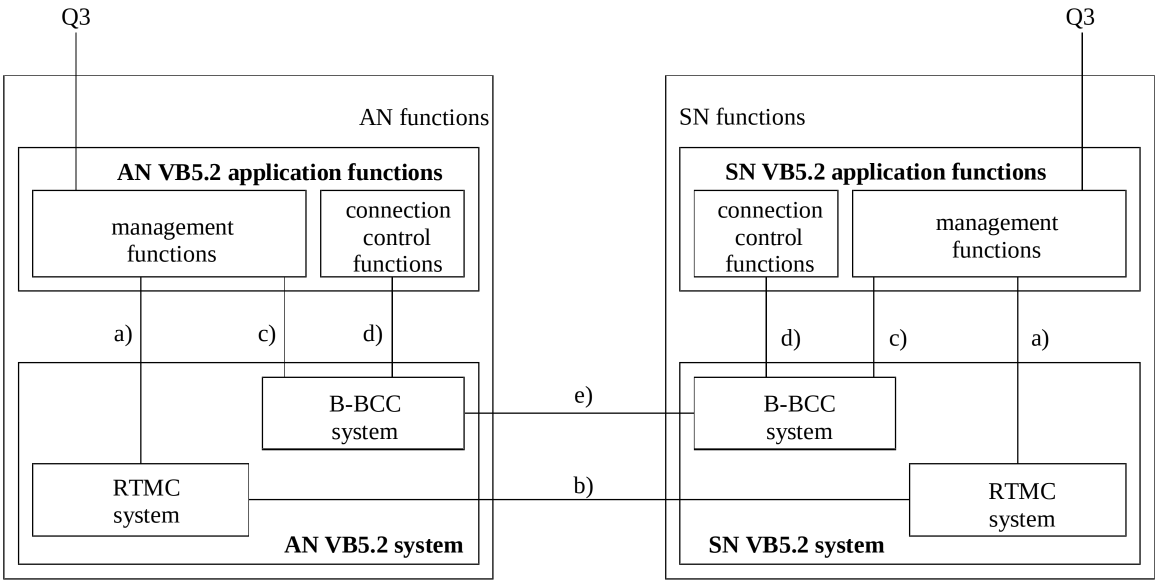
There is a requirement to be able to permit only test traffic across a VB5 interface.

Annex B (normative): Relationship between VB5.2 interfaces and the management model

B.1 Introduction

This annex describes the relationships between VB5.2 interfaces and the management model. In particular it describes when primitives (see "General functional architecture" [6]) are created due to messages from the OS and when messages are sent to the OS as a result of primitives generated by the managed system.

Figure B.1 shows the position of the management functions (which works on the management model and sends/receives OS messages) within a VB5.2 NE. The management function generates directly mee-Primitives to the VB5.2 system (channels a and c in the figure) and it triggers the connection control function to send cee-primitives to the VB5.2 system (channel d in the figure) for definition of mee- and cee-primitives see ITU-T Recommendations G.967.1 [5] and G.967.2 [6]. The interworking between management function and connection control function is not described on primitive level but just verbally.



a) Service primitives between management functions and RTMC system
b) RTMC protocol messages across the VB5.2 reference point
c) Service primitives between management functions and B-BCC system
d) Service primitives between connection control functions and B-BCC system
e) B-BCC protocol messages across the VB5.2 reference point

Figure B.1: Overall VB5.2 system specification model

In cases where attributes are changed as a result of primitives generated by the managed system, the OS may be informed by change notifications.

The following clauses within this annex describe the use of the various VB5 labels and VB5.2 procedures and relate the information model of the AN and the SN to the primitives of the VB5.2 system.

B.2 LSP, LUP and VPCI labels

The LSP identifier which is used in VB5 messages corresponds to the logicalServicePortNumber attribute of the logical service port objects. The LUP identifiers which are used in VB5 messages correspond to the logicalUserPort attribute of the VB5 UNI access objects in the SN or of the logical user port objects in the AN.

The VPCI values for LUPs in AN used in VB5 messages correspond to the values associated with the vpCtpAndVpciPtrList attribute of logical user port objects if the VPCs at the LUPs do not terminate in the AN, or to the values associated with the vpTtpAndVpciPtrList attribute of logical user port objects if the VPCs at the LUPs do terminate in the AN.

The VPCI values for LUPs in the SN used in VB5 messages correspond to the values associated with the tpAndVpciPtrList attribute of the VB5 UNI access objects for VPCs which terminate in the SN, or to the values associated with the vpCtpVb5AndVpciPtrList for VPCs which do not terminate in the SN. For VPCs which are associated with VB5 UNI accesses and which terminate in the AN, the VPCI values used in VB5 messages correspond to the values of the vpcLupNumber attribute of the vpcLup objects in the SN.

The VPCI values for LSPs used in VB5 messages correspond to the values associated with the vpTtpAndVpciPtrList attribute of the logical service port objects.

B.3 Shutting down

Shutting down is initiated by the OS of the AN changing the administrativeState attribute of an object which affects a VP or group of VPs related to the VB5 interface to its shutting-down value, or the partialAdministrativeState attribute to the partial shutting-down value for those objects which support this value. This results in the creation of a MEE_await_clear_req primitive or primitives in the AN.

Following the exchange of VB5 messages, the SN generates a MEE_await_clear_ind primitive or primitives which results in the changing of the remoteBlockingVb5 attribute from remoteUnblocked to remoteAwaitingClear in the relevant VB5 VP CTP or TTP objects or in the relevant vpcLup objects.

The SN responds to the MEE_await_clear primitive or primitives by waiting for calls to clear. When this is complete, the SN generates a MEE_await_clear_res primitive or primitives and sends the appropriate message to the AN, which responds and generates a MEE_await_clear_conf primitive or primitives. This allows the administrativeState or partialAdministrativeState attribute which initiated the process in the AN to change to locked or partially locked respectively.

B.4 Blocking and unblocking

When the relevant administrativeState or partialAdministrativeState attributes in the AN change to locked or partially locked, either as a result of shutting down or due to direct intervention by the OS, a MEE_block_request primitive with an administrative cause is generated and a message is sent to the SN. On receipt of this message, a MEE_block_ind primitive is generated in the SN. In addition, in the relevant VP CTP, TTP, vpcLup or logical service port objects the remoteBlockingVb5 attribute changes to remoteBlocked and the administrative field of the remoteBlockingReasonVb5 attribute changes to administrative cause partial or full, depending on the nature of the blocking.

If there is a fault which affects a VP or group of VPs in the AN, then a MEE_block_request primitive with a fault cause is generated, a message is sent to the SN, and often there will be an operationalState attribute in an object in the AN which changes to disabled. On receipt of the message, a MEE_block_ind primitive is generated in the SN and in the relevant VP CTP, TTP, vpcLup or logical service port objects the remoteBlockingVb5 attribute changes to remoteBlocked and the fault field of the remoteBlockingReasonVb5 attribute changes to error.

When the administrativeState or partialAdministrativeState attribute in the AN is changed to unlocked by the OS or the fault condition is cleared, a MEE_unblock_req primitive is generated in the AN and a message is sent to the SN. On receipt of this message, a MEE_unblock_ind primitive is generated in the SN and in the relevant VP CTP, TTP, vpcLup or logical service port objects the remoteBlockingVb5 attribute changes to remoteUnblocked and the fault or administrative field of the remoteBlockingReasonVb5 attribute changes to none.

The administrative and fault fields in the remoteBlockingVb5 attribute are independent.

B.5 VPCI consistency checking

The CheckVpciConsistency action is initiated by the OS of the SN via Q3 and is only applicable to VPCs on a VB5 interface which terminate in the AN and are associated with an LSP. The SN environment is responsible for ensuring that there is no second CheckVpciConsistency initiated as long the first one is running. The VPC on which the CheckVpciConsistency action is performed has to be in the operational state enabled. When starting the action, the operator has to provide the CheckVpciConsistencyInformation. The environment of the SN creates a MEE_cons_check_req primitive and a VB5 message is sent across the interface to the AN.

On receipt of this VB5 message, the AN generates a MEE_cons_check_ind primitive to activate the loopback monitoring function on the requested VPCI in AN environment. A MEE_cons_check_res primitive generated in the AN environment directed to the system management contains the information whether the activation of the loopback monitoring function was successful or the CheckVpciConsistency was rejected (e.g. if another CheckVpciConsistency started by a different SN is already running).

The appropriate VB5 message carries the result information back to the SN side. A MEE_cons_check_conf primitive is generated which triggers the SN environment to start sending end-to-end loopback cells (successful case) or leads to an action reply which is sent to the operator and terminates the CheckVpciConsistency action with the RemoteReason "notPerformed" (rejected or unknown resource case).

If the CheckVpciConsistency is successful up to this point, the detection by the SN of cells which have been looped back or the termination of the test results in the generation of a MEE_cons_check_end_req primitive followed by a VB5 message across the VB5.1 interface towards the AN.

On receipt of this VB5 message, the AN generates a MEE_cons_check_end_ind primitive which results in the deactivation of the loopback monitoring function. The AN environment generates a MEE_cons_check_end_res primitive and a VB5 message crosses the VB5.1 interface to the SN.

This message confirms the stopping of the VPCI consistency check procedure and carries the information whether the AN monitored the loopback cells or not (successful or failed). In the SN a MEE_cons_check_end_conf primitive is generated which transfers the result (successful, failed at AN) to the SN environment. The action reply CheckVpciConsistencyResult transfers this information via Q3 to the OS which started the action.

B.6 Interface start-up

The B-BCC start-up provides the communication procedures between SN and AN to initialize the B-BCC protocol entities by setting them to the idle state.

The startupLsp action is normally sent by the operator when no protocol is active (when the operationalState attribute of the related lspVb52 object class has the value "disabled"). The interface start-up procedure is initiated by a single startupLsp action and should occur in the following sequence:

a) RTMC start-up: refer to clause C.6 of EN 301 271 [1] but note that the lspActivationState attribute does not change to the value "activated" and no start-up action reply is sent after the RTMC start-up is finished.

b) B-BCC start-up: specified in this clause.

A B-BCC start-up shall be initiated by the NE systems after the RTMC protocol is running without failure. If only one protocol is active, the lspActivationState does not the have value "activated". From the view of the management information model, the B-BCC start-up involves the MOCs lspVb52Sn and lspVb52An with their startupLsp action and lspActivationState and operationalState attributes.

Values of the attribute lspActivationState are "activated", "restarting" and "not activated".

Values of the attribute operationalState are "enabled" and "disabled".

On the VB5.2 interface side the following primitives are related to the B-BCC start-up procedure:

- meeBbccStartupReq (SN side)
- meeBbccStartupConf (SN side)
- meeBbccRestartReq (SN side)
- meeBbccRestartConf (SN side)
- meeBbccStartResetInd (SN side)
- meeBbccStartTrafficInd (SN and AN side)
- meeBbccStartTrafficReq (AN side)
- meeBbccStartTrafficConf (AN side)
- meeBbccPresyncInd (AN side)
- meeBbccPresyncAccRes (AN side)
- meeBbccPresyncRejRes (AN side)

The following options for a start-up procedures are valid:

- B-BCC start-up SN side (see clause B.6.1);
- B-BCC start-up initiated by SN OS (see clause B.6.1.1);
- B-BCC start-up initiated autonomously by the SN NE (see clause B.6.1.2);
- B-BCC start-up from AN side (see clause B.6.2);
- B-BCC start-up initiated by AN OS (see clause B.6.2.1);
- Automatically initiated by the SN NE on receipt of an indication that the SAAL for the B-BCC protocol has been established by the AN (called later: B-BCC start-up automatically initiated by the AN NE) (see clause B.6.2.2).

Details of state transitions are given in the relevant state transition tables of annex C. Primitives in the SN and in the AN are linked through VB5 messages as described in the VB5 interface standards.

In the diagrams ' - A' indicates an action, ' - R' indicates an action reply, ' - N' indicates a notification and '(...)' indicates a new value of an attribute. These diagrams do not show all of the possible information flows.

B.6.1 B-BCC start-up from SN side

B.6.1.1 B-BCC start-up initiated by SN OS

SN events

The OS of the SN initiates an interface start-up through the startupLsp action of the lspVb52Sn object class. The action causes the management function to examine the value of the operationalState attribute. If the value is "enabled", the startupLsp action ends immediately and informs the OS of the SN through the start-up action reply.

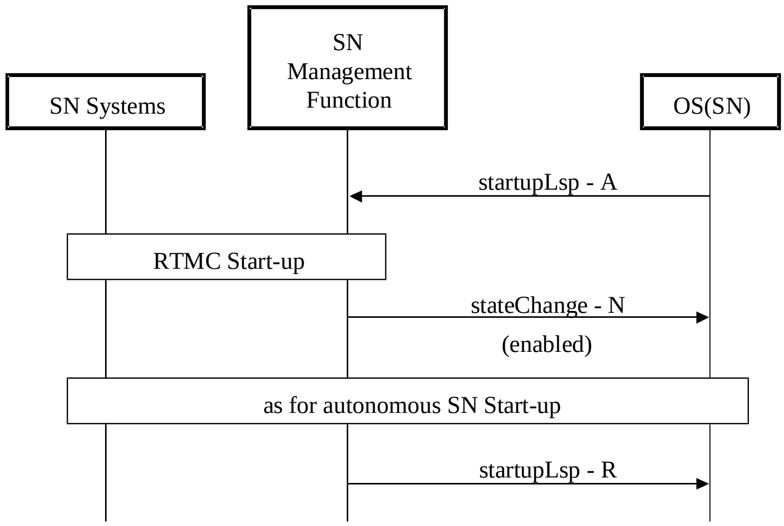
If the value of the operationalState attribute is "disabled", then the RTMC start-up is initiated, the value is of the changed to "enabled", and the management function of the SN generates a meeBbccStartupReq primitive towards the SN B-BCC system.

After completion of the procedure, the SN B-BCC system generates a meeBbccStartupConf primitive and the result information is presented to the OS with the action reply StartupLspResult.

The information syntax of this action reply reflects the addressed resource and could have at this stage the following values: "not successful" and "successful".

If the value of the lspActivationState attribute changes, an attributeValueChange notification is sent to the OS but if the state of the operationalState attribute changes, a stateChange notification is sent to the OS.

These information flows are illustrated in figure B.2.



NOTE: The SN events for the autonomous SN Start-up are shown in figure B.4.

Figure B.2: Interface start-up by OS(SN) - SN events

AN events

The AN management function receives a meeBbccStartTrafficInd primitive from the AN B-BCC system. The AN OS is informed by an attributeValueChange notification of the lspActivationState attribute.

The start-up procedure is completed by a reset. The AN connection control function informs the management function of the AN about a reset request from the SN which causes a resetBbccResult notification to be generated by the appropriate lspVb52An object. The information syntax of this notification reflects the addressed resource and the success/no success of the B-BCC reset procedure.

These information flows are illustrated in figure B.3.

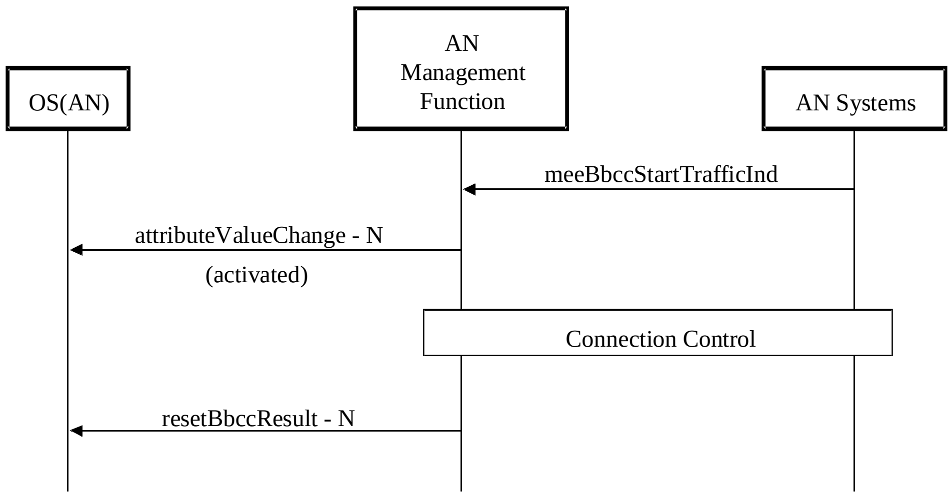


Figure B.3: Start-up initiated at SN side - AN events

B.6.1.2 B-BCC start-up initiated autonomously by the SN NE

SN events

In case of an automatic start-up procedure, the management function of the SN generates a meeBbccStartupReq primitive towards the SN B-BCC system. After completion of the procedure, the SN B-BCC system generates a meeBbccStartupConf primitive. If the value of the lspActivationState attribute changes, an attributeValueChange notification is sent to the OS but if the state of the operationalState attribute changes, a stateChange notification is sent to the OS.

These information flows are illustrated in figure B.4.

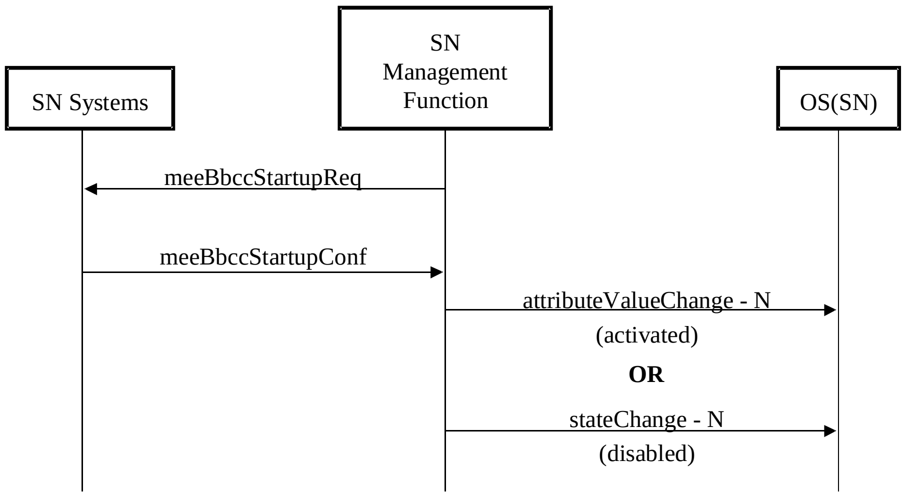


Figure B.4: Autonomous interface start-up by SN - SN events

AN events

Refer to the AN events for B-BCC start-up initiated by SN OS in clause B.6.1.1.

B.6.2 B-BCC start-up from AN side

B.6.2.1 B-BCC start-up initiated by AN OS

AN events

The OS of the AN initiates an interface start-up through the startupLsp action of the lspVb52Sn object class. The action causes the management function to examine the value of the operationalState attribute. If the value is "enabled", the startupLsp action ends immediately and informs the OS of the AN through the start-up action reply.

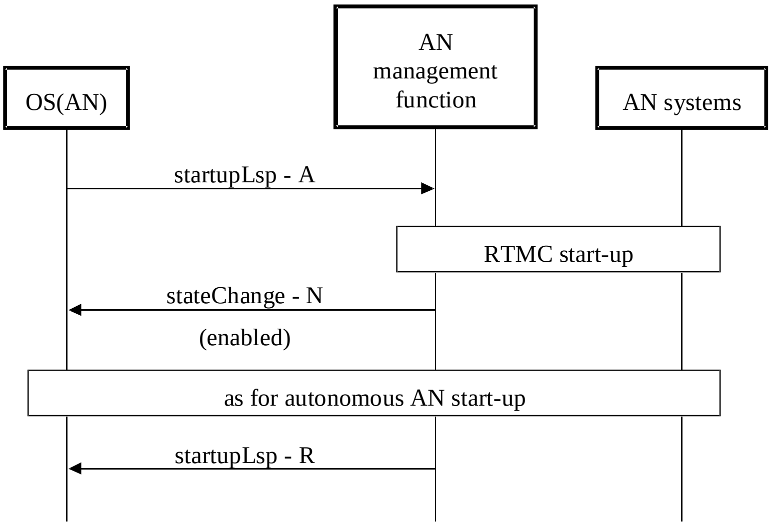
If the value of the operationalState attribute is "disabled", then the RTMC start-up is initiated, the value is of the changed to "enabled", and the management function of the AN generates a meeBbccStartupReq primitive towards the AN B-BCC system. After an establishment of the SAAL has taken place, the AN B-BCC system sends a meeBbccStartTrafficConf primitive to the AN management function.

The result information is presented to the OS with the action reply StartupLspResult. The information syntax of this action reply reflects the addressed resource and could have at this stage the following values: "not successful" and "successful".

If the value of the lspActivationState attribute changes, an attributeValueChange notification is sent to the OS but if the state of the operationalState attribute changes a stateChange notification is sent to the OS.

From this point on, refer to clause B.6.2.2.1 (if the SN side decides for a B-BCC start-up) or to clause B.6.2.2.2 (if the SN decides for a B-BCC Restart).

These information flows are illustrated in figure B.5.



NOTE: The AN events for the autonomous AN start-up are shown in figure B.6.

Figure B.5: Interface start-up by OS(AN) - AN events

SN events

Refer to the SN events for B-BCC start-up automatically initiated by the AN NE in clause B.6.2.2.

B.6.2.2 B-BCC start-up automatically initiated by the AN NE

AN events

In the case of an automatic start-up procedure, the management function of the AN generates a meeBbccStartTrafficReq primitive towards the AN B-BCC system.

After an establishment of the SAAL has taken place, the AN B-BCC system sends a meeBbccStartTrafficConf primitive to the AN management function.

If the value of the lspActivationState attribute changes, an attributeValueChange notification is sent to the OS, but if the state of the operationalState attribute changes, a stateChange notification is sent to the OS.

These information flows are illustrated in figure B.6.

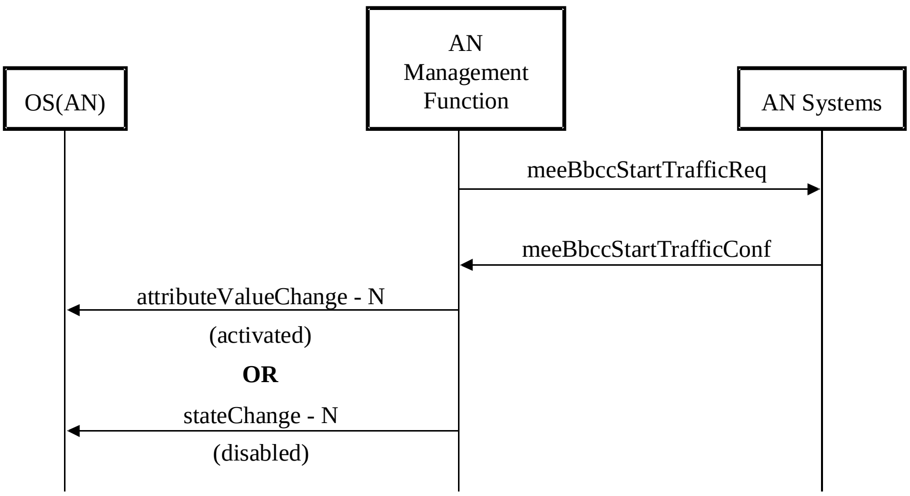


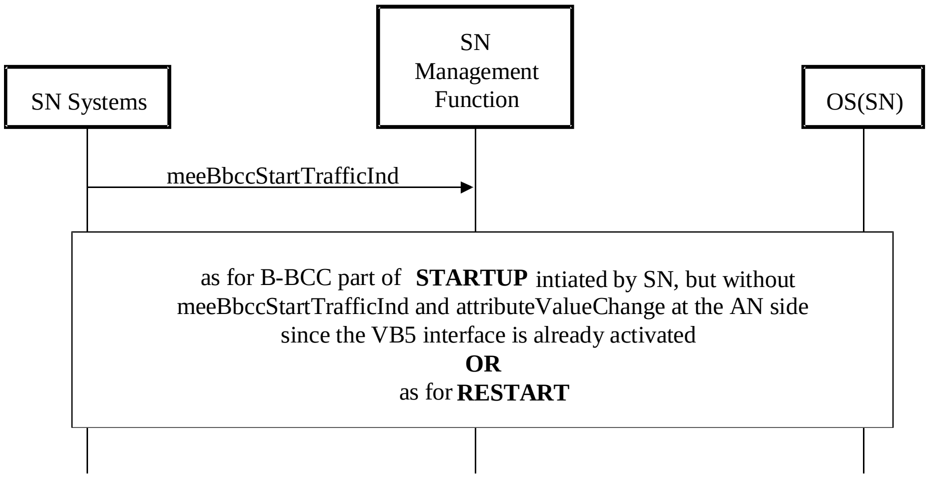
Figure B.6: Autonomous interface start-up by AN - AN events

SN events

On the SN side, the B-BCC system indicates with a meeBbccStartTrafficInd primitive that an SAAL establishment has taken place.

At this point the SN connection control function decides whether the SN proceeds with a B-BCC start-up as described in clause B.6.2.2.1 or a B-BCC restart as described in clause B.6.2.2.2. The SN OS is neither involved nor informed about this decision.

These information flows are illustrated in figure B.7.



NOTE: The STARTUP option is shown in figures B.3 and B.4.
The RESTART option is shown in figures B.8 and B.9.

Figure B.7: Start-up initiated at AN side - SN events

B.6.2.2.1 B-BCC start-up in SN due to automatic initiation of start-up in AN

SN events

Refer to the SN events for B-BCC start-up automatically initiated by the SN NE in clause B.6.1.2.

AN events

The AN connection control function informs the management function of the AN about a reset request from the SN which causes a resetBbccResult notification to be generated by the appropriate lspVb52An object. The information syntax of this notification reflects the addressed resource and the success/no success of the B-BCC reset procedure.

B.6.2.2.2 B-BCC restart in SN due to automatic initiation of start-up in AN

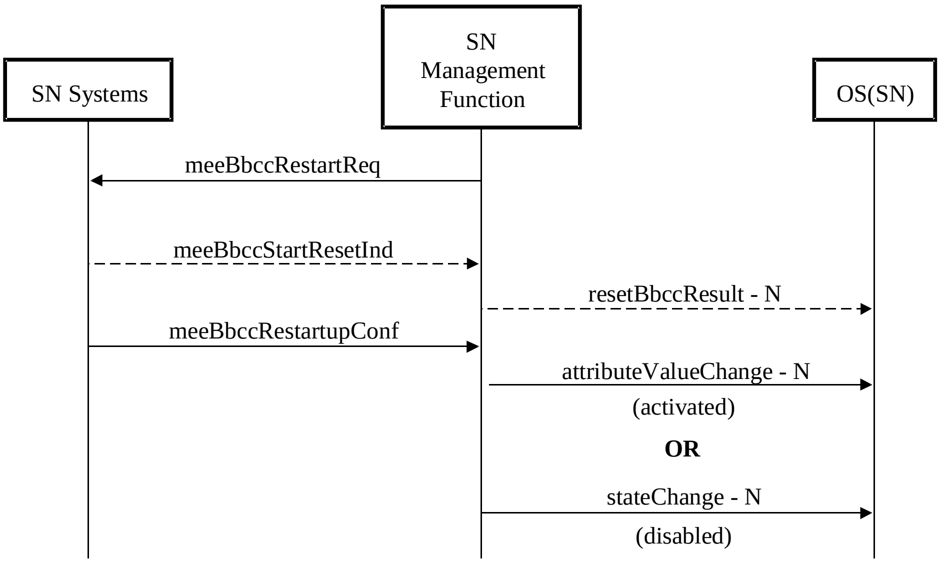
SN events

The SN management function responds with a meeBbccRestartReq primitive towards the SN B-BCC system. If the result of a presync is that a reset is necessary, the SN B-BCC system indicates this towards the SN management function by sending a meeBbccStartResetInd primitive. This causes a resetBbccResult notification to be generated by the appropriate lspVb52Sn object. The information syntax of this notification reflects the addressed resource and the success/no success of the B-BCC reset procedure.

After completion, the SN B-BCC system generates a meeBbccRestartConf primitive.

If the value of the lspActivationState attribute changes, an attributeValueChange notification is sent to the OS, but if the state of the operationalState attribute changes, a stateChange notification is sent to the OS.

These information flows are illustrated in figure B.8.



NOTE: The meeBbccStartResetInd primitive and the resetBbccResult notification only occur if the result of presynchronization process is that a reset is necessary.

Figure B.8: Start-up initiated at AN side - SN events for RESTART

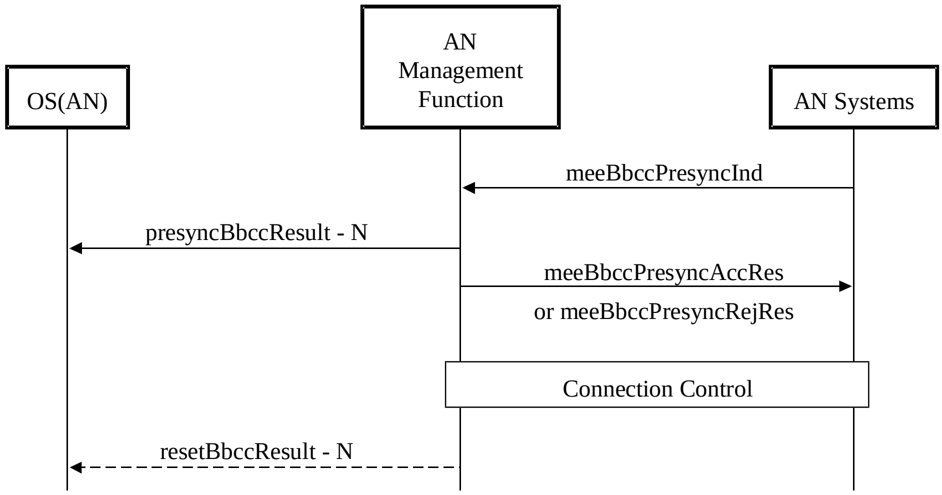
AN events

A meeBbccPresyncInd primitive, generated by the AN B-BCC system and sent towards the AN management function, indicates a request from the SN side to provide the information on whether the B-BCC protocol can resume its normal operation or not. If the B-BCC presync procedure is successful, the AN OS is informed by a presyncBbccResult notification.

A meeBbccPresyncAccRes primitive, sent from the AN management function to the AN B-BCC system, indicates that the B-BCC protocol can resume its normal operation and a B-BCC reset is not necessary.

Alternatively, a meeBbccPresyncRejRes primitive is sent from the AN management function to the AN B-BCC system to indicate that a B-BCC reset shall be initiated before the B-BCC protocol can resume its normal operation. In this case, existing connections over the interface will be lost. If a reset is necessary then the AN connection control function informs the management function of the AN about a reset request from the SN, which causes a resetBbccResult notification to be generated by the appropriate lspVb52An object. The information syntax of this notification reflects the addressed resource and the success/no success of the B-BCC reset procedure.

These information flows are illustrated in figure B.9.



NOTE: meeBbccPresyncAccRes means that no reset is necessary, whereas meeBbccPresyncRejRes means that a reset is necessary.

Figure B.9: Start-up initiated at AN side - AN events for RESTART

B.7 LSP identity checking

The check logical service port identification can be invoked from either side SN or AN. Due to a checkLspIdentification action initiated by an OS via the Q3 interface, the related environment creates a primitive MEE_verify_LSP_ID_req which results in the appropriate message across the VB5.1 interface.

On the other side, no MEE primitive is created to inform the environment about the procedure. A VB5 message is sent back to the SN containing the requested information about the logical service port identifier. After comparison of the two LSP Id values, a MEE_verify_LSP_ID_conf primitive is created which provides the result (positive result indication for consistency and negative result indication for mismatch) to the environment. The action reply checkLspIdentificationResult transfers the information via Q3 to the OS which started the action.

B.8 RTMC reset

The RTMC Reset procedure is carried out by the Logical Service Port managed object by means of the RTMC reset action. An RTMC reset results in the SN taking appropriate action which can include the release of on-demand connections although the intention is to minimize the interruption of service. Furthermore, the states of all VPCs and the state of the LSP are set to unblocked; VPCs not available for service due to administrative reasons shall be blocked again by the Logical Service Port managed object. Shut-down requests and VPCI consistency checks are aborted as a consequence of an RTMC reset request.

According to the interface specification, this procedure may be initiated both by the AN OS and by the SN OS and involves the peer system as well, where the procedure is activated by RTMC commands.

The case is described below; as the same primitives and managed objects are involved in the AN and the SN a generic description has been used; in particular, the managed object modelling the interface is called LSP and the VB5 System Management functional blocks in the AN and SN are called VB5 System Management.

The command sent by the OS will be carried on the Q3 interface by the RTMC reset action; the parameter specifies the managed object identifier that will carry it out.

The action command is received by MCF that will generate an internal message to the LSP managed object identified by the appropriate parameter; this message activates the RTMC reset action of the LSP that in turn will generate a MEE_reset_req to the VB5 System Management functional block.

The RTMC reset action is activated on the peer system by the RTMC VB5 messages across the VB5 interface; on receipt of the VB5 messages the VB5 System Management block of the remote system carries out the RTMC reset procedure and reports the result to the remote LSP managed object by means of a MEE_reset_ind primitive.

Only if the RTMC reset is triggered by the AN OS then the peer LSP in the SN, as soon as it receives the RTMC reset indication, informs the OS by the resetRtmcResult notification.

At the end of the VB5 messages phase the VB5 System Management block of the initiating system sends the LSP a MEE_reset_conf primitive with the result of the action, which may be successful or unsuccessful. Finally, the LSP managed object reports the result to the OS by the action reply.

The relationships described above are summarized in figure B.10; in this example, the AN is the initiating system.

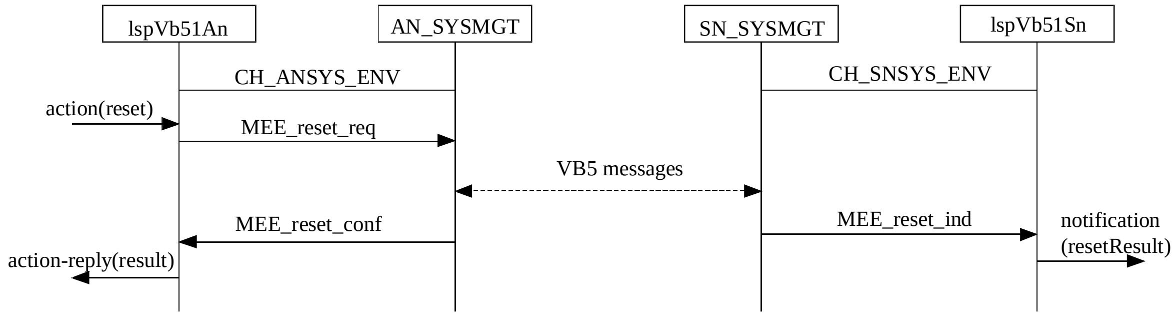


Figure B.10: RTMC Reset procedure triggered by the AN OS

B.9 B-BCC reset

A B-BCC reset may be initiated either by the OS of the SN or autonomously by the SN. If it is initiated by the OS, the B-BCC reset applies to the complete LSP.

The OS of the SN initiates a B-BCC reset through the resetBbcc action of the lspVb52Sn object class. The action causes the management function of the SN to trigger the connection control function to generate a ceeBbccResetReq primitive towards the B-BCC system. The success of the action is presented to the connection control function by the ceeBbccResetAccConf or ceeBbccResetRejConf primitive, given back to the management function and transported to the OS in the resetBbcc action reply syntax.

If a B-BCC reset is initiated autonomously by the SN (i.e. SN connection control function generates a ceeBbccResetReq primitive towards the B-BCC system), the management function in the SN is informed after the completion of the B-BCC reset procedure (i.e. connection control function receives a ceeBbccResetAccConf or ceeBbccResetRejConf primitive from the B-BCC system) and presented to the OS with the resetBbccResult notification. The information syntax of this notification reflects the addressed resource and the success/no success of the B-BCC reset procedure.

When the AN connection control function receives a ceeBbccResetInd primitive it informs the management function of the AN which causes a resetBbccResult notification to be generated by the appropriate lspVb52An object. The information syntax of this notification reflects the addressed resource and the success/no success of the B-BCC reset procedure.

B.10 Congestion

For further study.

Annex C (normative): State transitions

C.1 State transition tables for AN

Clause D.1 of EN 301 271 [1] applies with the following additions.

Table C.1 maps the transitions of the lspActivationState attribute and the operationalState attribute of MOC lspVb52An on MEE primitives towards the VB5.2 system in the AN. The same conventions as in clause D.1 of EN 301 271 [1] are applied.

Table C.1: Mapping of lspActivationState on MEE primitives in AN

<table>
<tr><th>OperationalState (note 1)</th><th colspan="3">enabled</th><th>disabled</th></tr>
<tr><th>lspActivationState (note 2)</th><th>activated</th><th>restarting</th><th>notActivated</th><th>notActivated</th></tr>
<tr><th>Event</th><th>1.1</th><th>1.2</th><th>1.3</th><th>2</th></tr>
<tr><td colspan="5">Q3 Actions</td></tr>
<tr><td>start-up LSP</td><td>Q3reply (successful);
-</td><td>Q3reply (restarting);
-</td><td>Q3reply (activating);
-</td><td>meeStartupReq;
1.3</td></tr>
<tr><td colspan="5">RTMC relevance (note 3)</td></tr>
<tr><td rowspan="2">meeStartupConf (success)</td><td rowspan="2">/</td><td>-;
1.1</td><td rowspan="2">meeBbccStartTrafficReq
;
-</td><td rowspan="2">/</td></tr>
<tr><td>-;
-</td></tr>
<tr><td rowspan="2">meeStartupConf
(no success)</td><td rowspan="2">/</td><td>meeStartupReq;
-</td><td rowspan="2">Q3reply (unsuccessful);
2</td><td rowspan="2">/</td></tr>
<tr><td>-;
2</td></tr>
<tr><td>meeStartupInd</td><td>-;
-</td><td>-;
-</td><td>-;
-</td><td>-;
1.3</td></tr>
<tr><td>meeLspFailureInd</td><td>meeStartupReq;
1.2</td><td>meeStartupReq;
-</td><td>meeStartupReq;
1.2</td><td>-;
-</td></tr>
<tr><td colspan="5">B-BCC relevance (note 4)</td></tr>
<tr><td rowspan="2">meeBbccStartTrafficConf
(success)</td><td rowspan="2">/</td><td>-;
1.1</td><td rowspan="2">Q3reply (successful);
1.1</td><td rowspan="2">/</td></tr>
<tr><td>-;
-</td></tr>
<tr><td rowspan="2">meeBbccStartTrafficConf
(no success)</td><td rowspan="2">/</td><td>meeBbccStartTrafficReq
;
-</td><td rowspan="2">Q3reply (unsuccessful);
2</td><td rowspan="2">/</td></tr>
<tr><td>-;
2</td></tr>
<tr><td>meeBbccStartTrafficInd</td><td>-;
-</td><td>-;
-</td><td>-;
1.1</td><td>/</td></tr>
<tr><td>meeBbccStopTrafficInd</td><td>meeBbccStartTrafficR
eq;
1.2</td><td>meeBbccStartTrafficReq
;
-</td><td>meeBbccStartTrafficReq
;
1.2</td><td>-;
-</td></tr>
<tr><td colspan="5">The following conventions are used:
< primitive [(attributes)] | Q3 action > ; < new state >
- no primitive or Q3 action to be generated or no state change
/ event not possible or not allowed for this state
NOTE 1: State change notifications are emitted for the operationalState attribute.
NOTE 2: Attribute value change notifications are emitted for the lspActivationState attribute.
NOTE 3: If two boxes are present in a single row, the upper one is valid if the B-BBC protocol is active; the lower one is valid if the B-BBC protocol is inactive.
NOTE 4: If two boxes are present in a single row, the upper one is valid if the RTMC protocol is active; the lower one is valid if the RTMC protocol is inactive.</td></tr>
</table>

C.2 State transition tables for SN

Clause D.2 of EN 301 271 [1] applies with the following addition.

Table C.2 maps the transitions of the lspActivationState attribute and the operationalState attribute of MOC lspVb52Sn on MEE primitives towards the VB5.2 system in the SN. The same conventions as in clause D.1 of EN 301 271 [1] are applied.

Table C.2: Mapping of lspActivationState on MEE primitives in SN

<table>
<tr><th>OperationalState (note 1)</th><th colspan="3">enabled</th><th>disabled</th></tr>
<tr><th>lspActivationState (note 2)</th><th>activated</th><th>restarting</th><th>notActivated</th><th>notActivated</th></tr>
<tr><th>Event</th><th>1.1</th><th>1.2</th><th>1.3</th><th>2</th></tr>
<tr><td colspan="5">Q3 Actions</td></tr>
<tr><td>start-up LSP</td><td>Q3reply (successful);
-</td><td>Q3reply (restarting);
-</td><td>Q3reply (activating);
-</td><td>meeStartupReq;
1.3</td></tr>
<tr><td colspan="5">RTMC relevance (note 3)</td></tr>
<tr><td rowspan="2">meeStartupConf (success)</td><td rowspan="2">/</td><td>-;
1.1</td><td rowspan="2">meeBbccRestartReq/meeBbccStartupReq (note 5);
-</td><td rowspan="2">/</td></tr>
<tr><td>-;
-</td></tr>
<tr><td rowspan="2">meeStartupConf (no success)</td><td rowspan="2">/</td><td>meeStartupReq;
-</td><td rowspan="2">Q3reply (unsuccessful);
2</td><td rowspan="2">/</td></tr>
<tr><td>-;
2</td></tr>
<tr><td>meeStartupInd</td><td>-;
-</td><td>-;
-</td><td>-;
-</td><td>-;
1.3</td></tr>
<tr><td>meeLspFailureInd</td><td>meeStartupReq;
1.2</td><td>meeStartupReq;
-</td><td>meeStartupReq;
1.2</td><td>-;
-</td></tr>
<tr><td colspan="5">B-BCC relevance (note 4)</td></tr>
<tr><td rowspan="2">meeBbccStartupConf (success)</td><td rowspan="2">/</td><td>-;
1.1</td><td rowspan="2">Q3reply (successful) (note 6);
1.1</td><td rowspan="2">/</td></tr>
<tr><td>-;
-</td></tr>
<tr><td rowspan="2">meeBbccStartupConf (no success)</td><td rowspan="2">/</td><td>meeBbccStartupReq;
-</td><td rowspan="2">Q3reply (unsuccessful) (note 6);
2</td><td rowspan="2">/</td></tr>
<tr><td>-;
2</td></tr>
<tr><td rowspan="2">meeBbccRestartConf (success)</td><td rowspan="2">/</td><td>-;
1.1</td><td rowspan="2">Q3reply (successful) (note 6);
1.1</td><td rowspan="2">/</td></tr>
<tr><td>-;
-</td></tr>
<tr><td rowspan="2">meeBbccRestartConf (no success)</td><td rowspan="2">/</td><td>meeBbccStartupReq;
-</td><td rowspan="2">Q3reply (unsuccessful) (note 6);
2</td><td rowspan="2">/</td></tr>
<tr><td>-;
2</td></tr>
<tr><td>meeBbccStartTrafficInd</td><td>-;</td><td>-;</td><td>meeBbccRestartReq/meeBbccStartupReq (note 5);
-</td><td>/</td></tr>
<tr><td>meeBbccStopTrafficInd</td><td>meeBbccRestartReq/meeBbccStartupReq (note 5);
1.2</td><td>meeBbccRestartReq/meeBbccStartupReq (note 5);
-</td><td>meeBbccRestartReq/ meeBbccStartupReq (note 5);
1.2</td><td>-;
-</td></tr>
<tr><td colspan="5">The following conventions are used:
< primitive [(attributes)] | Q3 action > ; < new state >
- no primitive or Q3 action to be generated or no state change
/ event not possible or not allowed for this state
NOTE 1: State change notifications are emitted for the operationalState attribute.
NOTE 2: Attribute value change notifications are emitted for the lspActivationState attribute.
NOTE 3: If two boxes are present in a single row, the upper one is valid if the B-BBC protocol is active; the lower one is valid if the B-BBC protocol is inactive.
NOTE 4: If two boxes are present in a single row, the upper one is valid if the RTMC protocol is active; the lower one is valid</td></tr>
</table>

OperationalState (note 1)	enabled			disabled
IspActivationState (note 2)	activated	restarting	notActivated	notActivated
Event	1.1	1.2	1.3	2

if the RTMC protocol is inactive.

NOTE 5: The decision which of meeBbccRestartReq and meeBbccStartupReq is sent is taken with the help of the connection control function.

NOTE 6: A Q3reply is only to be sent, if a Q3Action was received originally.

Annex D:
Void

Annex E (informative): Instantiation example

E.1 Conventions

The following conventions are used in the example (figure E.1):

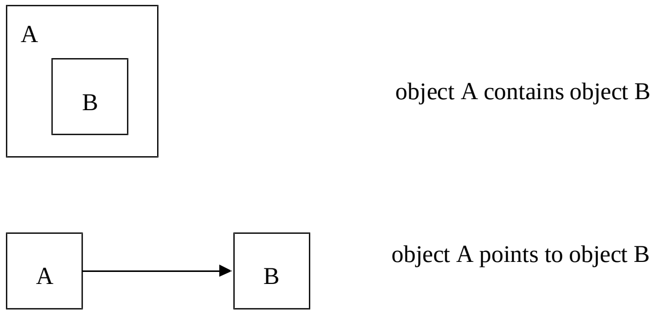


Figure E.1

E.2 Example of VPs/VCs allocation at the AN and the SN

In the example in figure E.2, it is supposed that there are two users, User#1 and User#2; User#1 has one VPC with two user VCCs allocated at the PUP, User 2 has two VPCs with one user VCC each. At the VB5.2 interface there are three VPs: VP 4 and VP 5 are used to carry user information, VP 6 contains two VCCs, one for the RTMC protocol and the other for the B-BCC protocol. The AN is a VP/VC cross connect; users' VCCs are semi-permanent VC connections created by provisioning.

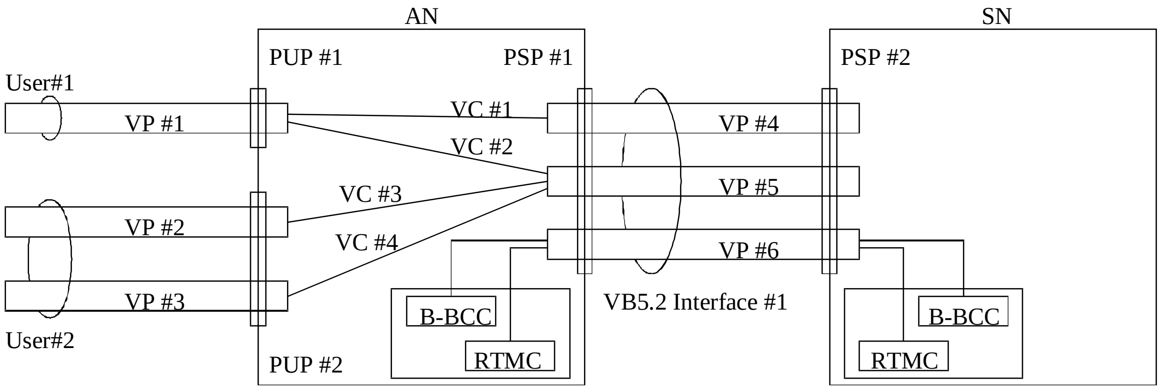


Figure E.2

E.3 Instantiation of managed objects in the AN

Figure E.3 shows the managed objects that are instantiated in the AN to model the VB5.2 interface and the VPs/VCs associated to the interface.

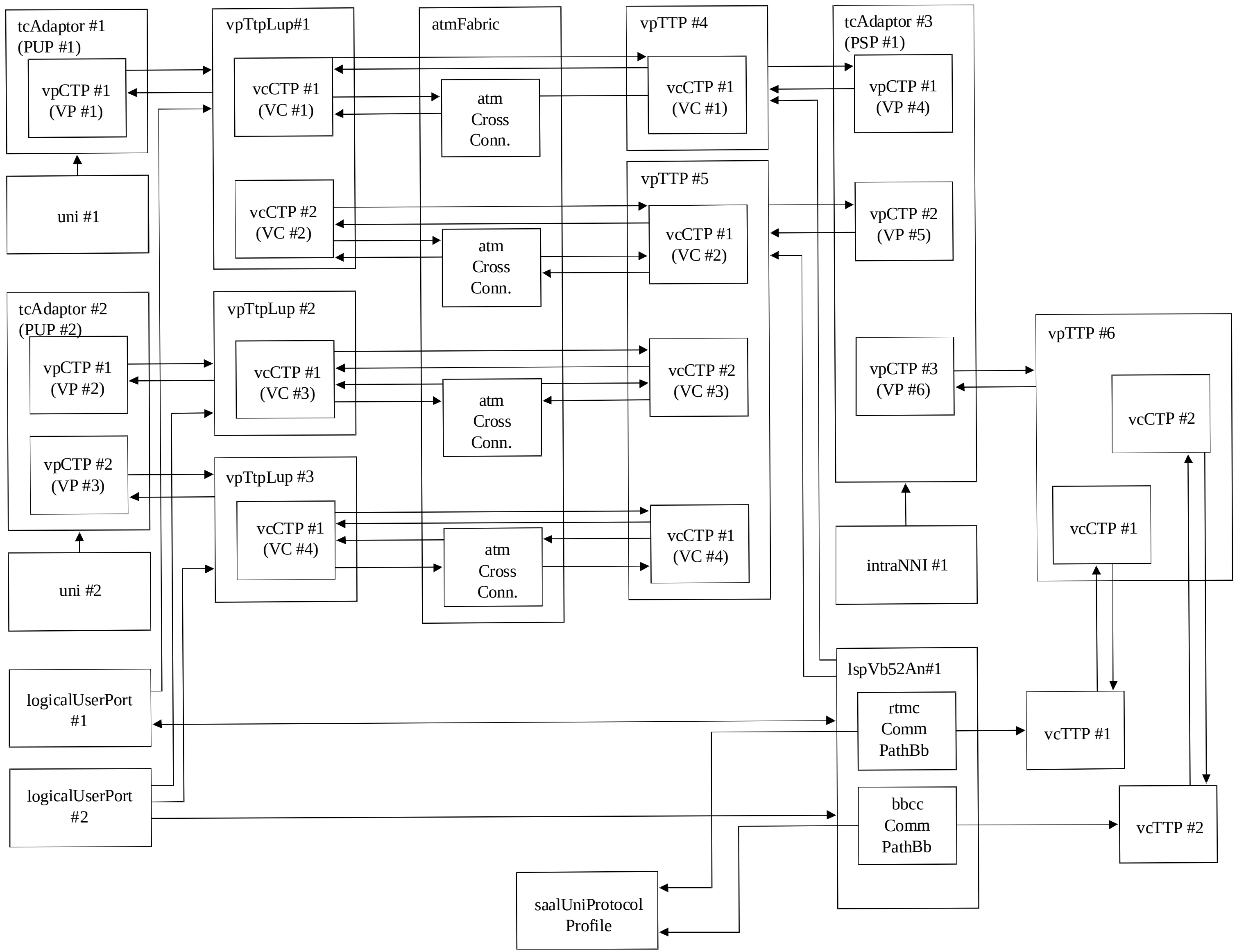

tcAdaptor #1
(PUP #1)
vpCTP #1
(VP #1)
uni #1
vpTtpLup#1
vcCTP #1
(VC #1)
vcCTP #2
(VC #2)
atmFabric
atm
Cross
Conn.
vpTTP #4
vcCTP #1
(VC #1)
vpTTP #5
vcCTP #1
(VC #2)
vcCTP #2
(VC #3)
vcCTP #1
(VC #4)
tcAdaptor #3
(PSP #1)
vpCTP #1
(VP #4)
vpCTP #2
(VP #5)
vpCTP #3
(VP #6)
intraNNI #1
vpTTP #6
vcCTP #2
vcCTP #1
tcAdaptor #2
(PUP #2)
vpCTP #1
(VP #2)
vpCTP #2
(VP #3)
uni #2
vpTtpLup #2
vcCTP #1
(VC #3)
vpTtpLup #3
vcCTP #1
(VC #4)
logicalUserPort
#1
logicalUserPort
#2
lspVb52An#1
rtmc
Comm
PathBb
bbcc
Comm
PathBb
vcTTP #1
vcTTP #2
saalUniProtocol
Profile

Figure E.3

The managed objects that have been introduced in addition to the VB5.1 managed objects to model the VB5.2 interface are the following:

- lspVb52An: it models the VB5.2 interface at the AN;
- bbccCommPathBb, it models the VCC used for the B-BCC protocol.

The atmAccessProfile is not drawn in the example, since it need not be instantiated.

E.4 Instantiation of managed objects at the SN

Figure E.4 shows the managed objects that are instantiated in the SN to model the VB5.2 interface and the VPs/VCs associated to the interface.

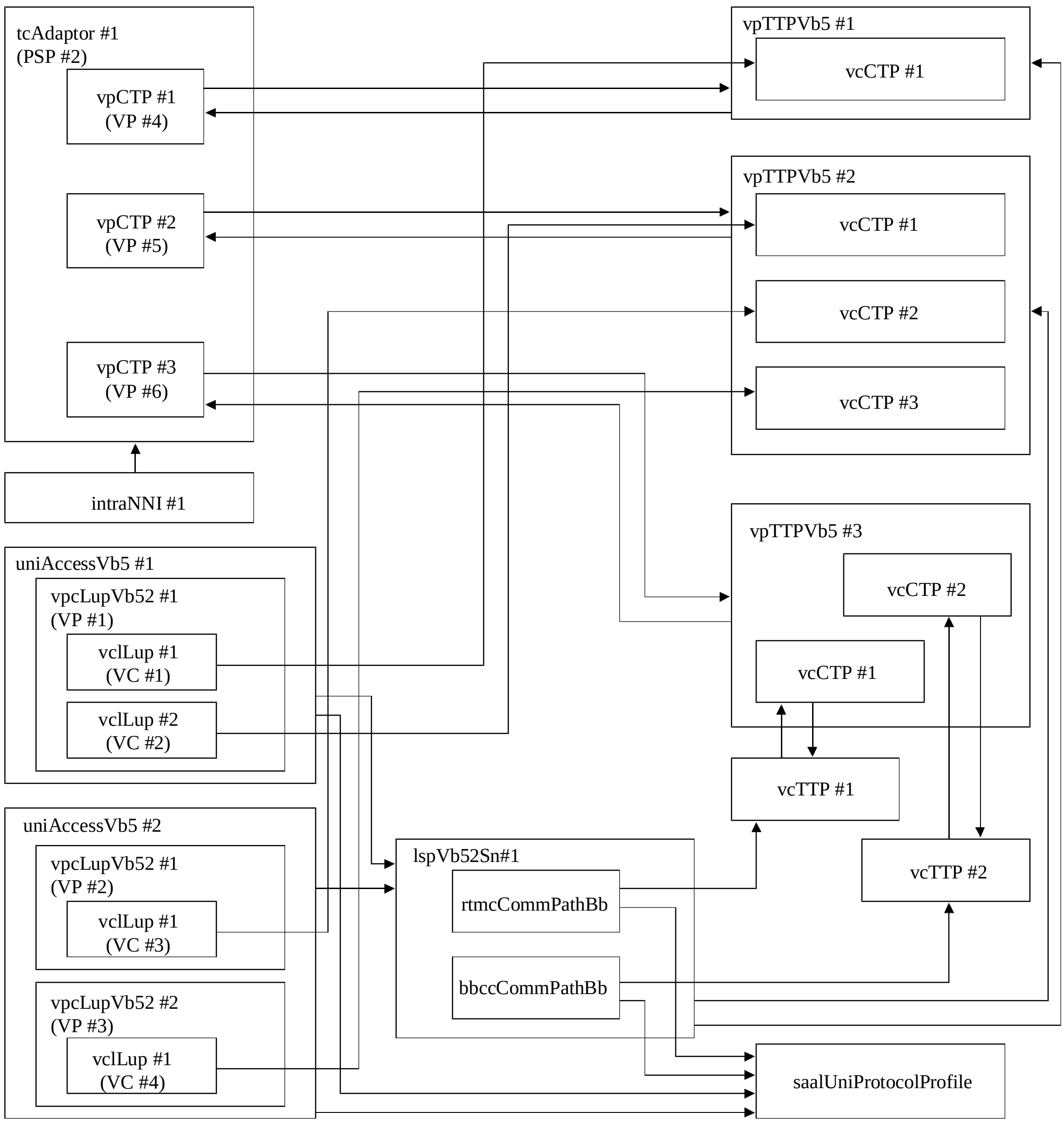


Figure E.4

The managed objects that have been introduced in addition to the VB5.1 managed objects to model the VB5.2 interface are the following:

- lspVb52Sn: it models the VB5.2 interface at the SN;
- bbccCommPathBb, it models the VCC used for the B-BCC protocol;
- vpcLupVb52, it models the VPs at the UNI that belong to the LUP, and that contain VCCs associated to a VB5.2 interface;
- vclLup, it models the individual VC-links at the UNI.

In this example, all objects point to the same saalUniProtocolProfile; in other cases, the objects may be associated with different saalUniProtocolProfile. Furthermore, the atmAccessProfile is not drawn, since it need not be instantiated.

Annex F (informative): Referenced Definitions

This annex contains the referenced GDMO and ASN.1 definitions from ITU-T Recommendation Q.832.2 [15], with known corrections included.

F.1 Object classes

This clause specifies the object classes for all of the managed objects used in the management information model. These object classes are either defined here or by reference to other specifications. Classes of managed objects which are defined elsewhere and which are only used for containment are not included, but are identified by the name bindings for the classes specified here.

Unidirectional trails are modelled by bidirectional objects with the traffic descriptor in the unused direction set to a null value.

The following classes which are defined in ITU-T Recommendation Q.832.1 [14] may be instantiated:

- logicalUserPort;
- rtmcCommPath;
- tcAdaptorTtpExtension;
- uniAccessVb5;
- vpcLup;
- vpCtpBidirectionalVb5;
- vpCtpBidirectionalVb5.

F.1.1 Profiling notes for imported classes

F.1.1.1 vpcLup (VPC at logical user port)

To avoid the deletion and re-creation of objects, instances of this class are used for the case of semi-permanent connections without the intention of including on-demand connections at a later date, only. Instances of vpcLupVb52 should be used otherwise.

F.1.2 Definition of classes

F.1.2.1 bbccCommPathBb (B-BCC communications path for broadband)

```
bbccCommPathBb MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Q.832.1":commPathBb;
    CHARACTERIZED BY
        bbccCommPathBbPkg PACKAGE
            BEHAVIOUR bbccCommPathBbBeh;;;
REGISTERED AS {q832-2ManagedObjectClass 1};

bbccCommPathBbBeh BEHAVIOUR
    DEFINED AS
        "The B-BCC communication path object class carries the B-BCC protocol information. Only one
object of this class shall be contained within the superior managed object";
```

F.1.2.2 lspVb52An (logical service port for VB5.2 in the access network)

```
lspVb52An MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Q.832.1":lspVb51An;
    CHARACTERIZED BY
        lspVb52AnPkg PACKAGE
            BEHAVIOUR lspVb52AnBeh;
            NOTIFICATIONS
                resetBbccResult,
                presyncBbccResult;;;
REGISTERED AS {q832-2ManagedObjectClass 2};

lspVb52AnBeh BEHAVIOUR
    DEFINED AS
        "This managed object represents a group of VP's coming from the same Service Node and
controlled by the same VB5.2 protocol.
        The resetBbccResult notification shall be emitted by the AN when a BBCC reset procedure is
initiated by the SN.
        The presyncBbccResult notification shall be emitted by the AN when a BBCC presynchronization
procedure is initiated by the SN";
```

F.1.2.3 lspVb52Sn (logical service port for VB5.2 in the service node)

```
lspVb52Sn MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Q.832.1":lspVb51Sn;
    CHARACTERIZED BY
        lspVb52SnPkg PACKAGE
            BEHAVIOUR lspVb52SnBeh;
            ATTRIBUTES
                connectionIdentifierFlag
                    DEFAULT VALUE Q832-2ASN1Module.connectionIdentifierFlagDefault
                    GET-REPLACE;
            ACTIONS
                resetBbcc;
            NOTIFICATIONS
                resetBbccResult
                ;;;
REGISTERED AS {q832-2ManagedObjectClass 3};

lspVb52SnBeh BEHAVIOUR
    DEFINED AS
        "This managed object represents a group of VP's coming from the same Access Network and
controlled by the same VB5.2 protocol.
        The connectionIdentifierFlag attribute indicates whether an exclusive procedure, when the
Service Node requests a dedicated VPCI/VCI combination, or a non-exclusive procedure, when the
Service Node proposes a preferred VPCI/VCI combination, is used for the selection of all on-demand
VPCI/VCI combinations on the Logical Service Port.
        The resetBbcc action initiates the BBCC reset procedure.
        The resetBbccResult notification shall be emitted by the SN when a BBCC reset procedure is
initiated autonomously by the NE.
        The presyncBbccResult notification shall be emitted by the SN when a BBCC presynchronization
procedure is initiated autonomously by the NE";
```

F.1.2.4 vclLup (VC link at the logical user port)

```
vclLup MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T X.721":top;
    CHARACTERIZED BY
        vclLupPkg PACKAGE
            BEHAVIOUR vclLupBeh;
            ATTRIBUTES
                vclLupId
                    GET
                    SET-BY-CREATE,
                bbccRequired
                    GET
                    SET-BY-CREATE,
                vciAtLup
                    GET
                    SET-BY-CREATE,
                vcCtpPtr
                    GET
                    SET-BY-CREATE;
```

```
            NOTIFICATIONS
                "ITU-T X.721": objectDeletion
            ;;;
REGISTERED AS {q832-2ManagedObjectClass 4};

vclLupBeh BEHAVIOUR
    DEFINED AS
        "The vclLup managed object class is a class of managed objects that provides the SN with
additional AN information about (semi-)permanent VC connections.
        A vclLup instance may be contained in a vpcLupVb52 instance only if the vpType attribute of
vpcLupVb52 is set to mixed.
        If the bbccRequired attribute is set to TRUE, the SN triggers the B-BCC protocol to
establish the VC connection in the AN.
        An object deletion notification is emitted, when an object instance is deleted automatically
as a consequence of the deletion of the associated vcCTPBidirectional instance";
```

F.1.2.5 vpcLupVb52 (VPC at the logical user port for VB5.2)

```
vpcLupVb52 MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T Q.832.1":vpcLup;
    CHARACTERIZED BY
        "ITU-T I.751":egressTrafficDescriptorPkg,
        "ITU-T I.751":ingressTrafficDescriptorPkg,
        "ITU-T I.751":qosClassesPkg,
        lupVcLevelProfilePkg ,
        vpcLupVb52Pkg PACKAGE
            BEHAVIOUR vpcLupVb52Beh;
            ATTRIBUTES
                "ITU-T Q.824.6":vpType
                    GET-REPLACE;;;
REGISTERED AS {q832-2ManagedObjectClass 5};

vpcLupVb52Beh BEHAVIOUR
    DEFINED AS
        "Each instance of the vpcLupVb52 managed object class models within the SN a single VPC that
belongs to an LUP and is terminated at the AN; within these VPCs shall be allocated only VCCs
carried towards the SN across a VB5.2 interface. Instances of this managed object class provide the
SN with the relevant information to select at the UNI a VPCs that may contain the VCC requested by
the user.
        An instance of vpcLupVb52 may contain semi-permanent or on-demand or both types of VC
connections; the class of the VC connections contained in the VPC is specified by the vpType
attribute.
        The packages egressTrafficDescriptorPkg and ingressTrafficDescriptorPkg provide information
concerning the traffic characteristic of the VP in both directions, ingress and egress; the
attribute values of these packages shall be provided to the managing system of the SN by the
managing system of the AN.
        The package qosClassesPkg contains the QoS class of the VP. The attribute values of this
package shall be provided to the managing system of the SN by the managing system of the AN.
        The package lupVcLevelProfilePkg provides the SN with the information concerning the remote
VC level profiling; the attribute values of this package shall be provided to the managing system of
the SN by the managing system of the AN";
```

F.1.2.6 vpTtpLup (VP trail termination point at the logical user port)

```
vpTtpLup MANAGED OBJECT CLASS
    DERIVED FROM "ITU-T I.751":vpTTPBidirectional;
    CHARACTERIZED BY
        "ITU-T Q.824.6": propagationDelayPkg,
        vpTtpLupPkg PACKAGE
            BEHAVIOUR vpTtpLupBeh;;;
REGISTERED AS {q832-2ManagedObjectClass 6};

vpTtpLupBeh BEHAVIOUR
    DEFINED AS
        "Each instance of the vpTtpLup managed object class models (within the AN) a single VPC that
belongs to an LUP and is terminated at the AN";
```

F.2 Name bindings

F.2.1 vclLup-vpcLupVb52

```
vclLup-vpcLupVb52 NAME BINDING
    SUBORDINATE OBJECT CLASS vclLup
        AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS vpcLupVb52
        AND SUBCLASSES;
    WITH ATTRIBUTE vclLupId;
    BEHAVIOUR vclLup-vpcLupVb52Beh;
    CREATE
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {q832-2NameBinding 1};

vclLup-vpcLupVb52Beh BEHAVIOUR
    DEFINED AS
        "An instance of a vclLup managed object class shall be deleted automatically when the
associated vcCTPBidirectional object instance (referred to by the vcCtpPtr) is deleted";
```

F.2.2 vpcLupVb52-uniAccessVb5

```
vpcLupVb52-uniAccessVb5 NAME BINDING
    SUBORDINATE OBJECT CLASS vpcLupVb52
        AND SUBCLASSES;
    NAMED BY SUPERIOR OBJECT CLASS "ITU-T Q.832.1":uniAccessVb5
        AND SUBCLASSES;
    WITH ATTRIBUTE "ITU-T Q.832.1":vpcLupId;
    BEHAVIOUR vpcLupVb52-uniAccessVb5Beh;
    CREATE
        WITH-AUTOMATIC-INSTANCE-NAMING;
    DELETE
        ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {q832-2NameBinding 2};

vpcLupVb52-uniAccessVb5Beh BEHAVIOUR
    DEFINED AS
        "An instance of vpcLupVb52 of vpType 'on-demand' or 'mixed' may not exist within a
uniAccessVb5 instance, when the latter is associated to a vpTTPBidirectionalVB5 instance of vpType
'on-demand' or 'mixed' (by the tpAndVpciSigPtrList)";
```

F.3 Definition of packages

F.3.1 lupVcLevelProfilePkg (Logical user port VC Level profile package)

```
lupVcLevelProfilePkg PACKAGE
    BEHAVIOUR lupVcLevelProfilePkgBeh;
    ATTRIBUTES
        "ITU-T I.751":maxNumVCIBitsSupported
            GET-REPLACE,
        "ITU-T I.751":maxNumActiveVCCsAllowed
            GET-REPLACE;
REGISTERED AS {q832-2Package 1};

lupVcLevelProfilePkgBeh BEHAVIOUR
    DEFINED AS
        "This package contains attributes that model the VC level profiling concerning a VPC of a
LUP associated to a VB5.2 interface. The attributes reflect the lower of the values supported by the
AN and the user. E.g. if the AN value is 8 and the user value is 6 then the attribute value is 6,
and if the AN value is 8 and the user value is 9 then the attribute value is 8";
```

F.4 Definition of attributes

F.4.1 bbccRequired (B-BCC required)

```
bbccRequired ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Q832-2ASN1ModuleBoolean;
    MATCHES FOR EQUALITY;
    BEHAVIOUR bbccRequiredBeh;
REGISTERED AS {q832-2Attribute 1};

bbccRequiredBeh BEHAVIOUR
    DEFINED AS
        "This attribute defines whether a (semi-)permanent connection in the SN is to be established
in the AN by using the B-BCC protocol. Moreover, if this is the case, the SN will re-establish the
connection autonomously in case of a failure of the connection in the AN";
```

F.4.2 connectionIdentifierFlag (Connection identifier flag)

```
connectionIdentifierFlag ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Q832-2ASN1Module.ConnectionIdentifierFlag;
    MATCHES FOR EQUALITY;
    BEHAVIOUR connectionIdentifierFlagBeh;
REGISTERED AS {q832-2Attribute 2};

connectionIdentifierFlagBeh BEHAVIOUR
    DEFINED AS
        "This attribute indicates whether an exclusive procedure, when the Service Node requests a
dedicated VPCI/VCI combination, or a non-exclusive procedure, when the Service Node proposes a
preferred VPCI/VCI combination, is used for the selection of on-demand VPCI/VCI combinations";
```

F.4.3 vcCtpPtr (VC CTP pointer)

```
vcCtpPtr ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Q832-2ASN1Module.ObjectInstance;
    MATCHES FOR EQUALITY;
    BEHAVIOUR vcCtpPtrBeh;
REGISTERED AS {q832-2Attribute 3};

vcCtpPtrBeh BEHAVIOUR
    DEFINED AS
        "This entity identifies the virtual channel connection termination point associated with the
object instance";
```

F.4.4 vciAtLup (VCI at the logical user port)

```
vciAtLup ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Q832-2ASN1Module.VciValue;
    MATCHES FOR EQUALITY;
    BEHAVIOUR vciAtLupBeh;
REGISTERED AS {q832-2Attribute 4};

vciAtLupBeh BEHAVIOUR
    DEFINED AS
        "This attribute defines the VCI value at the user side of a (semi-)permanent VC connection";
```

F.4.5 vclLupId (VC link at the logical user port identifier)

```
vclLupId ATTRIBUTE
    WITH ATTRIBUTE SYNTAX Q832-2ASN1Module.NameType;
    MATCHES FOR EQUALITY;
    BEHAVIOUR vclLupIdBeh;
REGISTERED AS {q832-2Attribute 5};

vclLupIdBeh BEHAVIOUR
    DEFINED AS
        "This attribute is used for naming instances of the class vclLup";
```

F.5 Definition of actions

F.5.1 resetBbcc (reset BBCC protocol)

```
resetBbcc ACTION
    BEHAVIOUR resetBbccBeh;
    MODE CONFIRMED;
    WITH REPLY SYNTAX Q832-2ASN1Module.ResetBbccResult;
REGISTERED AS {q832-2Action 1};

resetBbccBeh BEHAVIOUR
    DEFINED AS
        "This action is used to initiate the BBCC reset procedure";
```

F.6 Definition of notifications

F.6.1 presyncBbccResult (result of presynchronization of BBCC protocol)

```
presyncBbccResult NOTIFICATION
    BEHAVIOUR presyncBbccResultBeh;
REGISTERED AS {q832-2Notification 1};

presyncBbccResultBeh BEHAVIOUR
    DEFINED AS
        "This notification indicates to the operator that a BBCC presynchronization procedure, which
was initiated by the SN, took place successfully";
```

F.6.2 resetBbccResult (result of reset of BBCC protocol)

```
resetBbccResult NOTIFICATION
    BEHAVIOUR resetBbccResultBeh;
    WITH INFORMATION SYNTAX Q832-2ASN1Module.ResetBbccNotificationResult;
REGISTERED AS {q832-2Notification 2};

resetBbccResultBeh BEHAVIOUR
    DEFINED AS
        "This notification indicates to the operator that a BBCC reset procedure was initiated
either by the remote side or autonomously by the NE. The resource affected by the reset procedure
(according to the VB5.2 interface standard) is specified by the ResetBbccNotificationResult";
```

F.7 Type definitions

```
Q832-2ASN1Module {
        itu-t(0) recommendation (0) q(17) q832(832) dot(127) vb52(2)
            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 { ITU-T recommendation m 3100
            informationModel(0) asn1Modules(2) asn1DefinedTypesModule(0)}

    ProbableCause
```

```
        FROM Attribute-ASN1Module {joint-iso- ITU-T ms(9)
            smi(3) part2(2) asn1Module(2) 1}    -- X.721

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

-- end of imports


-- start of object identifier definitions

q832-2InformationModel
    OBJECT IDENTIFIER ::= {itu-t(0) recommendation (0)
        q(17) q832(832) dot(127) vb52(2) informationModel(0)}
q832-2StandardSpecificExtension
    OBJECT IDENTIFIER ::= {q832-2InformationModel standardSpecificExtension(0)}
q832-2ManagedObjectClass
    OBJECT IDENTIFIER ::= {q832-2InformationModel managedObjectClass(3)}
q832-2Package
    OBJECT IDENTIFIER ::= {q832-2InformationModel package(4)}
q832-2NameBinding
    OBJECT IDENTIFIER ::= {q832-2InformationModel nameBinding(6)}
q832-2Attribute
    OBJECT IDENTIFIER ::= {q832-2InformationModel attribute(7)}
q832-2Action
    OBJECT IDENTIFIER ::= {q832-2InformationModel action(9)}
q832-2Notification
    OBJECT IDENTIFIER ::= {q832-2InformationModel notification(10)}

vb52ProbableCause
    OBJECT IDENTIFIER ::= {q832-2StandardSpecificExtension 0}

-- end of object identifier definitions

-- The value assignments for the
-- ProbableCause parameter of the
-- VB5.2 specific TMN communications alarm notification
-- are specified below

bbccProtocolError
ProbableCause ::= globalValue : {vb52ProbableCause 1}
bbccProtocolSyntaxError
ProbableCause ::= globalValue : {vb52ProbableCause 2}
bbccProtocolTimeOutError
ProbableCause ::= globalValue : {vb52ProbableCause 3}

-- other ASN.1 definitions in alphabetical order

Boolean ::= BOOLEAN

ConnectionIdentifierFlag ::= INTEGER {
    exclusiveVpciVciCombination (0),
    preferredVpciVciCombination (1) }

connectionIdentifierFlagDefault
    ConnectionIdentifierFlag ::= exclusiveVpciVciCombination

ResetBbccNotificationResult ::= SEQUENCE {
    resource        ObjectInstance,
    result          ResetBbccResult }

ResetBbccResult ::= ENUMERATED{
    notSuccessful (0),
    successful (1)}


END -- of Q832-2 ASN.1Module
```

Annex G (informative): Bibliography

ETSI EN 301 217-1: "V interfaces at the digital Service Node (SN); Interfaces at the VB5.2 reference point for the support of broadband or combined narrowband and broadband Access Networks (ANs); Part 1: Interface specification".

ETSI ETR 257: "V interfaces at the digital Service Node (SN); Identification of the applicability of existing protocol specifications for a VB5 reference point in an access arrangement with Access Networks (ANs)".

ETSI EN 301 005-1: "V interfaces at the digital Service Node (SN); Interfaces at the VB5.1 reference point for the support of broadband or combined narrowband and broadband Access Networks (ANs); Part 1: Interface specification".

ETSI EN 301 064-1: "Telecommunications Management Network (TMN); Information models and protocols for the management and control of the Asynchronous Transfer Mode (ATM) switching network element; Part 1: Q3 interface specification".

ITU-T Recommendation G.774 (1992): "Synchronous Digital Hierarchy (SDH) management information model for the network element viewl".

ITU-T Recommendation G.803 (1997): "Architecture of transport networks based on the Synchronous Digital Hierarchy (SDH)".

ITU-T Recommendation G.851.1 (1996): "Management of the transport network - Application of the RM-ODP framework".

ITU-T Recommendation I.211 (1993): "B-ISDN service aspects".

ITU-T Recommendation I.311 (1996): "B-ISDN general network aspects".

ITU-T Recommendation I.327 (1993): "B-ISDN functional architecture".

ITU-T Recommendation I.356 (1996): "B-ISDN ATM layer cell transfer performance".

ITU-T Recommendation I.371 (1996): "Traffic control and congestion control in B-ISDN".

ITU-T Recommendation I.413 (1993): "B-ISDN user-network interface".

ITU-T Recommendations I.432.x series: "B-ISDN user-network interface - Physical layer specification".

ITU-T Recommendation I.580 (1995): "General arrangements for interworking between B-ISDN and 64 kbit/s based ISDN".

ITU-T Recommendation I.610 (1999): "B-ISDN operation and maintenance principles and functions".

ITU-T Recommendation I.732 (1996): "Functional characteristics of ATM equipment".

ITU-T Recommendation M.3010 (1996): "Principles for a telecommunications management network ".

ITU-T Recommendation M.3200 (1997): "TMN management services and telecommunications managed areas: Overview".

ITU-T Recommendation M.3207.1 (1996): "TMN management service: Maintenance aspects of B-ISDN management".

ITU-T Recommendation M.3400 (1997): "TMN management functions".

ITU-T Recommendation M.3610 (1996): "Principles for applying the TMN concept to the management of B-ISDN".

ITU-T Recommendation Q.821 (1993): "Stage 2 and stage 3 description for the Q3 interface - Alarm surveillance".

ITU-T Recommendation Q.824.5: "Stage 2 and stage 3 description for the Q3 interface - Customer administration: Configuration management of V5 interface environments and associated customer profiles".

ITU-T Recommendation X.208 (1988): "Specification of Abstract Syntax Notification One (ASN.1)".

ITU-T Recommendation X.701 (1997) | ISO/IEC 10040 (1998): "Information technology -Open Systems Interconnection - System management overview".

ITU-T Recommendation X.720 | ISO/IEC 10165-1 (1992): "Information technology - Open systems interconnection - Structure of management information: Management information model".

ITU-T Recommendation X.722 (1992) | ISO/IEC 10165-4 (1992): "Information technology - Open Systems Interconnection - Structure of management information: Guidelines for the definition of managed objects".

Amendment 1 (1995) to Recommendation X.722: "Information technology - Open systems interconnection - Structure of management information: Guidelines for the definition of managed objects: Amendment 1: Set by create and component registration".

ITU-T Recommendation X.731 | ISO/IEC 10164-2 (1992): "Information technology - Open systems interconnection - Systems management: State management function".

ITU-T Recommendation X.732 | ISO/IEC 10164-3 (1992): "Information technology - Open systems interconnection - Systems management: Attributes for representing relationships".

ITU-T Recommendation X.733 (1992) | ISO/IEC 10164-4 (1992): "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".

ITU-T Recommendation X.734 (1992) | ISO/IEC 10164-5 (1993): "Information technology - Open Systems Interconnection - Systems Management: Event report management function".

ITU-T Recommendation X.735 (1992) | ISO/IEC 10164-6 (1993): "Information technology - Open Systems Interconnection - Systems Management: Log control function".

ITU-T Recommendation X.737 (1995) | ISO/IEC 10164-14 (1996): "Information technology - Open Systems Interconnection - Systems management: Confidence and diagnostic test categories".

ITU-T Recommendation X.738 (1993) | ISO/IEC 10164-13 (1995): "Information technology - Open Systems Interconnection - Systems Management: Summarization function".

ITU-T Recommendation X.739 (1993) | ISO/IEC 10164-11 (1994): "Information technology - Open Systems Interconnection - Systems Management: Metric objects and attributes".

ITU-T Recommendation X.745 (1993) | ISO/IEC 10164-12:1994: "Information technology - Open Systems Interconnection - Systems Management: Test management function".

ITU-T Recommendation X.746 (1995) | ISO/IEC 10164-15 (1995): "Information technology - Open Systems Interconnection - Systems Management: scheduling function".

ITU-T Recommendation Q.822 (1994): "Stage 1, stage 2 and stage 3 description for the Q3 interface - Performance management".

ATM Forum Specification: "M4 Interface Requirements and Logical MIB: ATM Network Element View, (V1.0)".

ATM Forum Specification: "CMIP Specification for the M4 Interface, (V1.0)".

ATM Forum Specification: "ATM User-Network Interface Specification, (V3.0)".

ATM Forum Specification: "ATM User-Network Interface Specification, (V3.1)".

ATM Forum Specification: "ATM User-Network Interface Specification, (V4.0)".

History

Document history			
V1.1.0	March 2001	One-step Approval Procedure	OAP 20010720: 2001-03-21 to 2001-07-20
V1.1.1	September 2001	Publication	