

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Common control interface for networked digital audio and video products –
Part 5-1: Transmission over networks – General**

**Interface de commande commune pour produits audio et vidéo numériques
connectés en réseau –
Partie 5-1: Transmission sur des réseaux – Généralités**

IECNORM.COM : Click to view the full PDF of IEC 62379-5-1:2014



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2014 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 55 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Common control interface for networked digital audio and video products –
Part 5-1: Transmission over networks – General**

**Interface de commande commune pour produits audio et vidéo numériques
connectés en réseau –
Partie 5-1: Transmission sur des réseaux – Généralités**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX



ICS 33.160; 35.100

ISBN 978-2-8322-1693-4

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	4
INTRODUCTION.....	6
1 Scope.....	7
2 Normative references	7
3 Terms, definitions and abbreviations	7
3.1 Terms and definitions.....	7
3.2 Abbreviations.....	7
4 Network service specifications.....	8
4.1 Service for live media	8
4.2 Service for management messages	8
5 MIB definitions applicable to all networks	8
5.1 General.....	8
5.2 Type definitions	8
5.3 Conceptual row type definitions	9
5.4 MIB object definitions.....	10
5.4.1 Network ports	10
5.4.2 List of media sources.....	12
5.4.3 List of live media destinations.....	14
6 Calls.....	19
6.1 List of destinations in end equipment	19
6.2 Connecting a flow	20
6.3 Terminating a flow	20
6.4 Maintaining calls.....	21
7 Status broadcasts.....	21
7.1 General.....	21
7.2 Coding and encapsulation of reports	22
7.3 Standard report groups	23
7.3.1 General	23
7.3.2 List of sources	23
7.3.3 List of destinations	23
Annex A (informative) Machine-readable block definitions.....	24
Annex B (informative) Machine-readable data formats	36
Annex C (informative) Support for future networks	39
C.1 General.....	39
C.2 Services provided by the network.....	39
C.3 Network ports, flows, and media streams	40
C.3.1 Calls and flows	40
C.3.2 Connectivity model	40
C.3.3 Privilege	40
C.3.4 Call identity	40
C.4 Control of routing	41
C.5 Scheduled calls	41
Bibliography.....	42
Table 1 – Managed objects for network ports.....	10

Table 2 – Managed objects conveying the list of sources	12
Table 3 – Managed objects conveying the list of destinations	15

IECNORM.COM : Click to view the full PDF of IEC 62379-5-1:2014

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**COMMON CONTROL INTERFACE FOR NETWORKED
DIGITAL AUDIO AND VIDEO PRODUCTS –****Part 5-1: Transmission over networks –
General**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

International Standard IEC 62379-5-1 has been prepared by technical area 4: Digital system interfaces and protocol of IEC technical committee 100: Audio, video and multimedia systems and equipment.

The text of this standard is based on the following documents:

CDV	Report on voting
100/2107/CDV	100/2304/RVC

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

A list of all parts in the IEC 62379 series, published under the general title *Common control interface for networked digital audio and video products*, can be found on the IEC website.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IECNORM.COM : Click to view the full PDF of IEC 62379-5-1:2014

INTRODUCTION

Structure of the family of standards

IEC 62379 specifies the common control Interface, a protocol for managing networked audiovisual equipment. The following parts exist or are planned:

- 1 General
- 2 Audio
- 3 Video
- 4 Data
- 5 Transmission over networks
- 6 Packet transfer service
- 7 Measurement

IEC 62379-1:2007, specifies aspects which are common to all equipment, and it includes an introduction to the common control interface.

IEC 62379-2:2008, IEC 62379-3 (under consideration) and IEC 62379-4 (under consideration) specify control of internal functions specific to equipment carrying particular types of live media. IEC 62379-4 refers to time-critical data such as commands to automation equipment, but not to packet data such as the control messages themselves.

IEC 62379-5 specifies control of transmission of these media over each individual network technology. It includes network specific management interfaces along with network specific control elements that integrate into the control framework.

IEC 62379-5-1, (this standard) specifies management of aspects which are common to all network technologies.

IEC 62379-5-2 specifies protocols which can be used between networking equipment to enable the setting up of calls which are routed across different networking technologies.

IEC 62379-5-3, onwards, specify management of aspects which are particular to individual networking technologies.

IEC 62379-6, specifies carriage of control and status messages and non-audiovisual data over transports that do not support audio and video, such as RS232 serial links, with (as for IEC 62379-5) a separate subpart for each technology.

IEC 62379-7 specifies aspects that are specific to the measurement of the service experienced by audio and video streams and in particular to the requirements of EBU ECN-IPM Measurements Group.

COMMON CONTROL INTERFACE FOR NETWORKED DIGITAL AUDIO AND VIDEO PRODUCTS –

Part 5-1: Transmission over networks – General

1 Scope

This part of IEC 62379 specifies aspects of the common control interface that are common to all network technologies, including setting up and tearing down of sessions and the service provided by the network.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 62379-1:2007, *Common control interface for networked digital audio and video products – Part 1: General*

IEC 62379-5-2:2014, *Common control interface for networked digital audio and video products – Part 5-2: Transmission over networks – Signalling*

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62379-1 and IEC 62379-5-2, as well as the following apply.

3.1.1

media port

source or destination of media data in an interface unit

Note 1 to entry: A media port is either a physical port (e.g. an external audio or video connector on the unit) or a logical port (e.g. an internal connection to another part of the unit).

3.1.2

switch

network element which routes media data and other messages between links

3.2 Abbreviations

TCP Transmission Control Protocol ^a

UDP User Datagram Protocol ^b

MIB Management Information Base

^a See RFC 793.

^b See RFC 768.

4 Network service specifications

4.1 Service for live media

Live media (including status broadcasts) shall be transmitted using a service for which, if the network supports it, guaranteed levels of throughput, delay, and data loss shall be requested.

4.2 Service for management messages

Management messages should be transmitted in data units on an asynchronous flow as specified in IEC 62379-5-2. If no such service is available, a connectionless datagram service such as UDP may be used.

Where a connection-oriented service is used, at least one call at each privilege level shall be accepted by a destination unit at any given time. If more calls at one privilege level are accepted, this shall not prevent the acceptance of at least one call at each other privilege level.

5 MIB definitions applicable to all networks

5.1 General

The structure of the MIB shall be as specified in IEC 62379-1.

5.2 Type definitions

The following application-wide types shall be used:

```
NetPortState ::= INTEGER {
    disabled          (1),
    closing           (2),
    linkDown          (3),
    linkUp            (4), -- does not know to what linked
    pointToPoint      (5), -- link is point-to-point
    peerGroup         (6), -- e.g. Ethernet hub
    sharedMedia       (7), -- with master, e.g. 802.11
} (disabled..sharedMedia)
```

```
PortIdentifier ::= OCTET STRING (SIZE(3))
' octet 1 = port type
' octets 2 and 3 = port number (high byte in octet 2)
```

```
ConnectionEnd ::= INTEGER {
    source          (1),
    destination     (2)
} (source..destination)
```

```
CauseCode ::= OBJECT IDENTIFIER
```

NOTE Cause codes may be defined in other parts of the IEC 62379-5 series or elsewhere.

```
ConnectionState ::= INTEGER {
    readyToConnect   (1),
    connectionRequested (2),
    terminating      (3),
    active           (4),
    failed           (5),
    disconnected      (6),
    pending          (7),
    inactive         (8),
    finished         (9),
}
```

```

        callProceeding      (10),
        receivedOffer       (11),
        acceptedOffer       (12),
        reservationRequested (13),
        clearing            (14)
    } (readyToConnect..clearing )

    Importance ::= INTEGER (1..255)

    Priority ::= INTEGER (1..255)

```

5.3 Conceptual row type definitions

The following types are used to specify the syntax of managed objects in this standard that represent conceptual table rows.

```

NetPortEntry ::= SEQUENCE {
    nPortBlockId      BlockId,
    nPortName         Utf8String,
    nPortState        NetPortState,
    nPortAddressType  TDomain,
    nPortAddress       TAddress,
    nPortPAddrType    TDomain,
    nPortPartnerAddr  TAddress,
    nPortBarred       TruthValue
}

UnitSourceEntry ::= SEQUENCE {
    usFlowIdentifier  OCTET STRING,
    usBlockId        SourceBlockId,
    usBlockInput      IndexNumber,
    usPackageSize     CardinalNumber,
    usPrivilege       PrivilegeLevel,
    usState           ConnectionState,
    usCause           CauseCode,
    usSource          Utf8String,
    usDestination     Utf8String,
    usService         Utf8String,
    usImportance      Importance,
    usPriority         Priority,
    usStartTime       DateTime,
    usEndTime         DateTime,
    usConnectTime     CardinalNumber,
    usFlowIdStandard  TruthValue
}

UnitDestEntry ::= SEQUENCE {
    udFlowIdentifier  OCTET STRING,
    udNetBlockId     SourceBlockId,
    udNetBlockOutput  IndexNumber,
    udSourceAddrType  TDomain,
    udSourceAddress   TAddress,
    udPackageSize     CardinalNumber,
    udPrivilege       PrivilegeLevel,
    udState           ConnectionState,
    udCause           CauseCode,
    udSource          Utf8String,
    udDestination     Utf8String,
    udService         Utf8String,
    udImportance      Importance,
    udPriority         Priority,

```

```

udStartTime      DateTime,
udEndTime        DateTime,
udConnectTime    CardinalNumber,
udConnectCount   CardinalNumber,
udRemembered     TruthValue,
udDestBlockId    DestBlockId,
udDestBlockInput IndexNumber,
usFlowIdStandard TruthValue
}

```

5.4 MIB object definitions

5.4.1 Network ports

5.4.1.1 General

Each physical connection to a network, in a switch or end equipment, shall be represented using a network port block. A network port block shall have one input for each outgoing media flow and one output for each incoming media flow.

The group of objects in Table 1 shall be implemented by all switches. The root node for these objects shall be

```
{ iso(1) standard(0) iec62379(62379) network(5) general (1) networkMIB(1) networkPorts(1) }
```

This node shall be used as the block type identifier for network port blocks.

Table 1 – Managed objects for network ports

Identifier	Syntax	Index	Readable	Writable	Volatile	Syntax
netPortTable(1)	SEQUENCE OF NetPortEntry		none	none	yes	mandatory
└netPortEntry(1)	NetPortEntry		none	none	yes	mandatory
└└nPortBlockId(1)	BlockId	yes	none	none	no	mandatory
└└nPortName(2)	Utf8String		listener	supervisor	no	mandatory
└└nPortState(3)	NetPortState		listener	supervisor	yes	mandatory
└└nPortAddressType(4)	TDomain		listener	none	no	mandatory
└└nPortAddress(5)	TAddress		listener	none	no	mandatory
└└nPortPAddrType(6)	TDomain		listener	none	yes	mandatory
└└nPortPartnerAddr(7)	TAddress		listener	none	yes	mandatory
└└nPortBarred(8)	TruthValue		listener	supervisor	no	mandatory

5.4.1.2 netPortTable

A table of network port descriptors for this unit. Each physical network port on the unit shall have a corresponding entry in this table. There may also be entries for "virtual" network ports.

5.4.1.3 netPortEntry

An entry in the network port table.

5.4.1.4 nPortBlockId

The block identifier for this port. Used as an index when accessing the network port table.

5.4.1.5 nPortName

The name assigned to this port. This is an arbitrary text string assigned by the system manager. Such assignment should persist across resets of the unit.

Until a name has been assigned, this object shall have a value that relates to a visible marking associated with the port's physical connector.

Note that the name of a port on a network switch whose connector is labelled "2" on the unit's enclosure should default to a value such as "Port 2" or "Ethernet port 2" or "Front panel port 2". If it is connected to a network socket in, say, studio 6, a supervisor may then rename it as "Studio 6".

5.4.1.6 nPortState

The current link-layer state of the port's network connection.

If a management terminal sets this object to `closing` or `linkDown`, the managed unit shall reroute or, if that is not possible, gracefully close down all calls that pass through the port. In the case of `closing`, the port shall then enter the `disabled` state.

For as long as the port is in `linkDown` state, the managed unit shall attempt to establish a network connection on the port.

5.4.1.7 nPortAddressType

The type of network address used for `nPortAddress`.

5.4.1.8 nPortAddress

An address which identifies the port.

NOTE This will normally be the 48-bit MAC address of the interface. An IP address may be used if it is permanently assigned, but not if it is acquired via DHCP.

5.4.1.9 nPortPartnerAddrType

The type of network address used for `nPortPartnerAddr`.

5.4.1.10 nPortPartnerAddr

In `pointToPoint` state, the address of the unit to which the port is connected.

In `sharedMedia` state, the address of the unit which controls the local network to which the port is connected, e.g. wireless base station or clock master.

In other states, the `nPortPartnerAddr` value is not defined by this standard.

NOTE 1 This object is intended to allow a management terminal to "crawl" a network to discover its topology and what resources are present. The address allows it to make a management connection to the link partner in the case of a point-to-point link, or, in the case of a shared-media network segment, to a unit which may be able to supply a list of all the units on the segment. In contrast to `nPortAddress`, the `nPortPartnerAddr` value should identify the unit rather than its interface, so an EUI-64 is appropriate.

NOTE 2 For some kinds of network segment, such as an Ethernet segment using CSMA/CD, there is no straightforward method for enumerating all the units present on the segment.

5.4.1.11 nPortBarred

`false` (the default) if the unit is allowed to connect a route via the port; `true` if forbidden.

5.4.2 List of media sources

5.4.2.1 General

The “list of sources” has an entry for each synchronous flow transmitted by the managed unit.

The group of objects in Table 2 shall be implemented by all end equipment that can be the source for a synchronous flow, and by all switches. The root node for these objects shall be

{ iso(1) standard(0) iec62379(62379) network(5) general(1) networkMIB(1) callSources(2) }

NOTE 1 Calls are always connected by the destination, so this table is read-only, apart from the ability to clear down a call from the source end. Management calls are not included in this list.

NOTE 2 It is assumed that incoming calls specify a source in some way that may be at least partially network-dependent, and whenever a new connection is made an entry appears in this table, disappearing again when the call is released.

Table 2 – Managed objects conveying the list of sources

Identifier	Syntax	Index	Readable	Writable	Volatile	Syntax
unitSourceListTable(1)	SEQUENCE OF UnitSourceEntry		none	none	yes	mandatory
└UnitSourceEntry(1)	UnitSourceEntry		none	none	yes	mandatory
└usFlowIdentifier(1)	OCTET STRING	yes	none	none	yes	mandatory
└usBlockId(2)	SourceBlockId	yes	listener	none	yes	mandatory
└usBlockInput(3)	IndexNumber		listener	none	yes	mandatory
└usPackageSize(4)	CardinalNumber		listener	none	yes	mandatory
└usPackageRate(5)	CardinalNumber		listener	none	yes	mandatory
└usPrivilege(6)	PrivilegeLevel		listener	none	yes	mandatory
└usState(7)	ConnectionState		listener	see 6.3	yes	mandatory
└usCause(8)	CauseCode		listener	see 6.3	yes	mandatory
└usSource(9)	Utf8String		listener	none	yes	mandatory
└usDestination(10)	Utf8String		listener	none	yes	mandatory
└usService(11)	Utf8String		listener	none	yes	mandatory
└usImportance(12)	Importance		listener	none	yes	mandatory
└usPriority(13)	Priority		listener	none	yes	mandatory
└usStartTime(14)	DateandTime		listener	none	yes	mandatory
└usEndTime(15)	DateandTime		listener	none	yes	mandatory
└usConnectTime(16)	CardinalNumber		listener	none	yes	mandatory
└usFlowIdStandard(17)	TruthValue		listener	none	yes	mandatory

5.4.2.2 unitSourceListTable

The list of flows for which this unit transmits media data.

NOTE In the case of end equipment, the table lists flows for which the unit is the source. In the case of a switch, it lists information relating to onwards transmission towards the destination(s), for all synchronous flows passing through the unit.

5.4.2.3 unitSourceEntry

An entry in the list of flows for which this unit transmits media data.

5.4.2.4 usFlowIdentifier

An octet string which identifies the flow. The format specified in IEC 62379-5-2 should be used if available; see 5.4.2.20.

5.4.2.5 usBlockId

The identifier of the network port block for the unit's network port through which this flow passes.

5.4.2.6 usBlockInput

The input number, to the network port identified by usBlockId, through which this flow passes.

NOTE The entry associated with this input in connectorTable (see Table 2 of IEC 62379-1:2007) shows the block output which is the source of the media stream. The entry associated with that output in modeTable (see Table 2 of IEC 62379-1:2007) shows the media format being transmitted.

5.4.2.7 usPackageSize

The maximum number of payload octets that may be transmitted in a single data unit on the flow.

NOTE This is the size negotiated for the flow, which, when multiplied by the usPackageRate value, defines the bandwidth required.

It is not the maximum transmission unit size for the links over which the flow will be transmitted. For example:

- for an ATM link this would be fixed at the value 48, the number of payload octets in a cell;
- for an RTP stream this would be the maximum payload size for the RTP packets.

5.4.2.8 usPackageRate

The number of data units per second that may be transmitted on the flow.

5.4.2.9 usPrivilege

The privilege level associated with this flow, which is the highest of the privilege levels associated with its destinations if the network provides that information, supervisor otherwise.

5.4.2.10 usState

The current state of this flow. The callProceeding state shall indicate that an incoming connection request has been accepted and confirmation from the caller is awaited.

5.4.2.11 usCause

This object shall be initialised to a zero-length value, which shall indicate “normal call clearing”, and shall be set by the managed unit when it changes usState to failed or disconnected. See also 6.3.

5.4.2.12 usSource

The source name for this flow.

NOTE In the case of end equipment, the source name is specified as part of the definition of one of the blocks through which the signal passes on its way to the network port. In the case of a switch, it is inherited from the flow's udSource object (see 5.4.3.16).

5.4.2.13 usDestination

The destination name for the most important destination of the flow reached via this network port.

5.4.2.14 usService

The service name for the most important destination of the flow reached via this network port.

5.4.2.15 usImportance

The importance of the most important destination of the flow reached via this network port.

5.4.2.16 usPriority

The priority of the part of the flow connected via this network port.

5.4.2.17 usStartTime

The time at which the state is expected to change from pending to reservationRequested or active.

5.4.2.18 usEndTime

The time after which the network may release the resources reserved for the flow on this port.

5.4.2.19 usConnectTime

The number of seconds for which this flow has been active on this network port or the maximum CardinalNumber value, whichever is the less; zero if the state is not active.

5.4.2.20 usFlowIdStandard

true if usFlowIdentifier is in the format specified in IEC 62379-5-2, false otherwise.

Note that if the format specified in IEC 62379-5-2 is used, a flow has the same identifier in every piece of equipment through which it passes. Thus, records in which usFlowIdStandard or (in the list of destinations, see 5.4.3.28) udFlowIdStandard is true, even in units managed differently, refer to the same flow if, and only if, they have the same usFlowIdentifier or udFlowIdentifier value.

Note that if usFlowIdStandard is false, the management terminal should regard the flow identifier simply as a value chosen by the managed unit which, when combined with usBlockId, forms an index to this table. Thus, the management terminal should not assume any relationship between flows on different ports in unitSourceListTable, or between flows in unitSourceListTable and in unitDestListTable (see 5.4.3), based on a comparison of their identifiers; relationships should be established via connectorTable (see Table 2 of IEC 62379-1:2007).

5.4.3 List of live media destinations

5.4.3.1 General

The “list of destinations” has an entry for each synchronous flow received by the managed unit.

The group of objects in Table 3 shall be implemented by all end equipment that can be the destination for synchronous flows, and by all switches. The root node for these objects shall be

{iso(1) standard(0) iec62379(62379) network(5) general(1) networkMIB(1) callDestinations(3)}

Table 3 – Managed objects conveying the list of destinations

Identifier	Syntax	Index	Readable	Writable	Volatiles ^a	Syntax
unitNextFlowId(1)	OCTET STRING		listener	none	yes	see ^b
unitNextCallId(2)	OCTET STRING		listener	none	yes	see 5.4.3.3
unitDestListTable(3)	SEQUENCE OF UnitDestEntry		none	none	yes	mandatory
└UnitDestEntry(1)	UnitDestEntry		none	none	yes	mandatory
└udFlowIdentifier(1)	OCTET STRING	yes	none	none	maybe	mandatory
└udNetBlockId(2)	SourceBlockId		listener	see 6.1	yes	mandatory
└udNetBlockOutput(3)	IndexNumber		listener	none	yes	mandatory
└udSourceAddrType(4)	TDomain		listener	see 6.1	maybe	mandatory
└udSourceAddress(5)	TAddress		listener	see 6.1	maybe	mandatory
└udPackageSize(6)	CardinalNumber		listener	none	maybe	mandatory
└udPackageRate(7)	CardinalNumber		listener	none	maybe	mandatory
└udPrivilege(8)	PrivilegeLevel		listener	see 6.1	maybe	mandatory
└udState(9)	ConnectionState		listener	see 6.1	yes	mandatory
└udCause(10)	CauseCode		listener	see 6.1	yes	mandatory
└udSource(11)	Utf8String		listener	none	maybe	mandatory
└udDestination(12)	Utf8String		listener	see 6.1	maybe	mandatory
└udService(13)	Utf8String		listener	see 6.1	maybe	mandatory
└udImportance(14)	Importance		listener	see 6.1	maybe	mandatory
└udPriority(15)	Priority		listener	see 6.1	maybe	mandatory
└udStartTime(16)	DateandTime		listener	see 6.1	maybe	mandatory
└udEndTime(17)	DateandTime		listener	see 6.1	maybe	mandatory
└udConnectTime(18)	CardinalNumber		listener	none	maybe	mandatory
└udConnectCount(19)	CardinalNumber		listener	supervisor	maybe	optional
└udRemembered(20)	TruthValue		listener	see 6.1	no	see ^b
└udDestBlockId(21)	DestBlockId		listener	see 6.1	maybe	see ^b
└udDestBlockInput(22)	IndexNumber		listener	see 6.1	maybe	see ^b
└udFlowIdStandard(23)	TruthValue		listener	none	maybe	mandatory
Management calls are not included in this list. Synchronous flows are always connected by the destination, so in the case of end equipment every entry in this list is a flow which has been initiated by a management terminal through the process detailed in Clause 6.						
^a Where the volatility is shown as "maybe", the object is volatile, if, and only if, udRemembered is false.						
^b Objects shown as "see ^b " are mandatory for end equipment but should return noSuchName in units that cannot be the destination of a media flow.						

5.4.3.2 unitNextFlowId

A flow identifier value, not in the form specified in IEC 62379-5-2, for which a new record is created in the table, as described in 6.1. Consecutive Get requests for this object shall return different values. Values should be chosen in a way that maximises the time before re-use of any particular value. A GetNext request for this object shall not create a new record. The value returned (if any) is not defined by this standard.

Whereas the port number is an index into the list of sources, the source flow numbers only need to be unique for each port, in the list of destinations the flow number is the only index, and therefore it needs to be unique across all ports of the managed unit.

Managed units that support flow identifiers in the format specified in IEC 62379-5-2 should internally allocate a call identifier as if `unitNextCallId` had been requested. The value returned may be the call identifier, or a flow identifier based on it, or some other value which refers to it. In any case, the record that is created shall have `udFlowIdStandard` set to `false`, and the value returned shall be different from any flow identifier in the format specified in IEC 62379-5-2 that could be created using a call identifier returned by reading `unitNextCallId`.

To meet this requirement, it is sufficient for the octet string to be a different length from the format specified in IEC 62379-5-2, or to begin with the internally allocated call identifier.

5.4.3.3 `unitNextCallId`

A new call identifier value in the form specified in IEC 62379-5-2, from which flow identifiers may be derived, as described in 6.1. Consecutive Get requests for this object shall return different values. Values should be chosen in a way that maximises the time before re-use of any particular value. The value (if any) returned for a GetNext request for this object is not defined by this standard.

This object is mandatory for end equipment that supports flow identifiers in the format specified in IEC 62379-5-2, but shall return `noSuchName` in units that do not support that format and in units that return `noSuchName` for `unitNextFlowId`.

5.4.3.4 `unitDestList`

The list of synchronous flows which this unit receives.

NOTE In the case of end equipment, the table lists flows for which the unit is the destination. In the case of a switch, it lists information relating to reception from the source, for all media flows passing through the unit.

5.4.3.5 `unitDestEntry`

An entry in the list of synchronous flows which this unit receives.

5.4.3.6 `udFlowIdentifier`

An octet string which identifies the flow uniquely within the unit. See also 5.4.3.28.

Note that this is the only index field, so it shall be unique within the unit, not merely for each port.

5.4.3.7 `udNetBlockId`

In `inactive` and `readyToConnect` states, the identifier of the network port block for the unit's network port through which this flow should be connected, or `nullBlock` if the network port is to be chosen by the managed unit. The managed unit may ignore this value even if it is not `nullBlock`.

In other states, the identifier of the network port block for the unit's network port through which this flow passes, or zero if the flow is not associated with a specific network port.

5.4.3.8 `udNetBlockOutput`

The output number, of the network port identified by `udNetBlockId`, through which this flow passes, or zero if `udNetBlockId` is zero or the output has not yet been chosen.

NOTE The entry associated with this output in `modeTable` (see Table 2 of IEC 62379-1:2007) shows the media format being received, and `connectorTable` (see Table 2 of IEC 62379-1:2007) shows to which block inputs (if any) the received media stream is being routed internally. In the case of a switch, these blocks will be the network ports on which the flow is being output.

5.4.3.9 udSourceAddrType

The type of network address used for `ucSourceAddress`.

5.4.3.10 udSourceAddress

The network address of the source.

5.4.3.11 udPackageSize

The maximum number of payload octets that may be transmitted in a single data unit on the flow.

NOTE See examples in 5.4.2.7.

5.4.3.12 udPackageRate

The number of data units per second that may be transmitted for the flow.

5.4.3.13 udPrivilege

The privilege level associated with this flow.

5.4.3.14 udState

The current state of this flow.

5.4.3.15 udCause

This object shall be initialised to a zero-length value, which shall indicate “normal call clearing”, and shall be set by the managed unit when it changes `udState` to `failed` or `disconnected`. See also 6.3.

5.4.3.16 udSource

The source name for this flow.

NOTE This object is not writable by the management terminal. The source name is inherited from the source of the flow.

5.4.3.17 udDestination

The destination name for this flow.

5.4.3.18 udService

The service name for this flow.

5.4.3.19 udImportance

The importance for this flow.

5.4.3.20 udPriority

The reconnection priority for this flow.

5.4.3.21 udStartTime

This object shall be initialised to a zero-length octet string. The managed unit may set it to the time at which the flow should be connected, see 6.2.

5.4.3.22 udEndTime

The time after which the network may release the resources reserved for this flow. A zero-length octet string indicates that the flow is to remain connected indefinitely.

5.4.3.23 udConnectTime

The number of seconds for which this flow has been active or the maximum `CardinalNumber` value, whichever is less, zero if the state is not active.

5.4.3.24 udConnectCount

The number of times this flow has been connected (includes both rerouting and automatic reconnection). Connection of a flow is only counted when it has been active for at least 1 min. The count may be reset by writing the value zero. Writing any other value shall be rejected with the `badValue` error code.

5.4.3.25 udRemembered

If `true`, indicates this is a “remembered” flow which should be reconnected after a disconnection by the network (for instance as a result of a link failure) or a reset of the managed unit.

5.4.3.26 udDestBlockId

When `udState` is `readyToConnect` or `active`, indicates the identifier of the block to which this flow should be internally connected, or `nullBlock` if no internal connection is to be made. The value in other states is not defined by this standard.

Note that there are two ways to connect a block's input to an external source. The input can be identified by `udDestBlockId` and `udDestBlockInput`, in which case the managed unit makes the internal connection, or these objects can be left as zero and the internal connection are made explicitly after the managed unit has set `udNetBlockOutput` to a non-zero value. The former is a simpler process for the management terminal. The latter allows the management terminal to check the format being received before making the internal connection and, if the destination block already has a connection, to wait until data units are being received on the new connection before making the switch.

5.4.3.27 udDestBlockInput

The input number, of the block identified by `udDestBlockId`, to which this flow should be internally connected.

The value of this object shall be ignored by the managed unit when connecting a flow if `udDestBlockId` is `nullBlock`, also if the block identified by `udDestBlockId` only has one input.

5.4.3.28 udFlowIdStandard

`true` if `udFlowIdentifier` is in the format specified in IEC 62379-5-2, `false` otherwise.

Note that

- if the format specified in IEC 62379-5-2 is used, a flow has the same identifier in every piece of equipment through which it passes. Thus, records in which `udFlowIdStandard` or (in the list of sources, see 5.4.2.20) `usFlowIdStandard` is `true`, even in units managed differently, refer to the same flow if, and only if, they have the same `udFlowIdentifier` or `usFlowIdentifier` value;
- if `udFlowIdStandard` is `false`, the management terminal should regard the flow identifier simply as a value chosen by the managed unit which forms an index to this table. Thus, the management terminal should not assume any relationship between flows on different ports in this table, or between flows in Table 2 and in the table specified in 5.4.2, based on a comparison of their identifiers. Relationships should be established via `connectorTable` (see Table 2 of IEC 62379-1:2007).

6 Calls

6.1 List of destinations in end equipment

When a unit sends a `GetResponse` which includes a `unitNextFlowId` value in reply to a `Get` request, it shall create a "new" entry in its `unitDestList` for that value with `udFlowIdStandard` set to `false`.

In this subclause, a "new" entry is one in which all numeric objects not otherwise specified are zero and all octet-string and object-identifier objects are zero-length, with `udPrivilege` set to the privilege of the sender of the `Get` request, `udState` `inactive`, and `udRemembered` `false`.

When a unit receives a `Set` request for a column in an entry in its `unitDestList` which does not exist, but where the index is a valid flow identifier in the form specified in IEC 62379-5-2, and includes a call identifier which has previously been returned as a reply to a `Get` request for `unitNextCallId`, it shall create a "new" entry in its `unitDestList` for that index with `udFlowIdStandard` set to `true`.

A management terminal shall not send a `Set` request which could create an entry as described in the preceding paragraph unless either it has established that the entry already exists or the index is a flow identifier in the form specified in IEC 62379-5-2 in which the call identifier value is one that it has previously received in reply to a `Get` request for `unitNextCallId`.

Note that if the managed unit receives a `Set` request with an index which could be an IEC 62379-5-2 flow identifier which includes a call identifier which has not previously been returned as a reply to a `Get` request for `unitNextCallId`, it should reply `noSuchName`. However, it may also create a new entry without checking the call identifier.

The managed unit shall delete an entry when its `udState` is set to `finished`, and may delete an entry for which `udRemembered` is `false` and `udState` is `failed` or `disconnected` or `inactive` after a time-out which shall be at least 5 min.

An end unit shall refuse a `Set` request for any of the objects whose "writable" privilege is shown in Table 3 as "see 6.1" unless the request has a privilege which is at least as high as the current `udPrivilege` value for the entry. In the case of objects other than `udState`, `udDestination`, `udService`, `udImportance`, and `udPriority`, the request shall also be refused if the current `udState` is not `inactive`. In the case of `udState`, the request shall also be refused unless it changes the value from `inactive` to `readyToConnect`, or from any value to `terminating`, or from `failed` to `inactive` or `readyToConnect`, or from `failed` or `disconnected` or `inactive` to `finished`.

NOTE 1 An operator, for example, can interfere with calls set up by other operators and by listeners but not with those set up by supervisors. It is assumed that an operator in one area will not have access to, and therefore not be able to interfere with, equipment in other areas.

In the case of a unit that cannot be the destination of a synchronous flow (such as a switch, unless it also has media ports), objects whose "writable" privilege is shown in Table 3 as "see 6.1" shall be read-only except as specified in 6.3.

NOTE 2 This means that flows can only be connected from the destination, not from an intermediate point in the route. It also means that flows can only be "remembered" at a destination.

6.2 Connecting a flow

A management terminal requiring to connect a flow shall first acquire an identifier for the flow, either by reading the destination unit's `unitNextFlowId` object or by constructing one from a call identifier issued to it by the destination unit. It shall then Set the objects in the `unitDestEntry` for that identifier to values appropriate to the flow.

In the last such Set request (if there is more than one) the management terminal shall set the `udState` to `readyToConnect`.

When `udState` is set to `readyToConnect`, if `udStartTime` is a zero-length octet string the managed unit shall attempt to connect the flow. During the connection process, `udState` may take other values such as `callProceeding`, depending on the signalling procedures used. These values may be further specified in IEC 62379-5-2 or an other part of the IEC 62379-5 series¹ which applies to the network used for the call.

If `udFlowIdStandard` is `true`, the procedures specified in IEC 62379-5-2 should be used. The initial request will be an `AddFlow` if the route is already connected, `FindRoute` otherwise. Otherwise, procedures specific to the network technology should be used. If the flow is successfully connected, the managed unit shall set the value of `udState` to `active`. If the request is refused, the unit shall set the value of `udState` to `failed` and the value of `udCause` to the cause code returned by the network. The management unit may then set the value of `udState` to `inactive` and then change any of the other objects in the `unitDestEntry` as appropriate to reattempt the connection (for instance, to try an alternative address). It may also change `udState` directly to `readyToConnect`, to reattempt with the same parameters. If the cause is indicated to be temporary, and `udState` is still `failed` after a minimum of $(256 - udPriority)/10$ s, then the managed unit shall reattempt the connection.

If `udStartTime` is a valid `DateTime` object when `udState` is set to `readyToConnect`, and if the network supports pre-reservation, the managed unit shall request that resources be reserved which will allow the flow to be connected at the indicated time. On successful completion of this process it shall set `udState` to `pending`. If the network does not support pre-reservation the managed unit shall set `udState` to `pending` immediately, and then attempt to connect the flow at the latest time such that if the flow is successfully connected the transition to `active` will occur before the indicated time.

6.3 Terminating a flow

When, as a result of a Set request, the `udState` of an entry in a unit's `unitDestList` is set to `terminating`, the managed unit shall request the network to disconnect the corresponding flow, supplying the value in `udCause` as the cause. On completion of the disconnection, the unit shall set `udState` to `failed` if an error was signalled by the network, otherwise to `disconnected`. If there is no corresponding flow (for instance, if the previous state was `inactive`) the unit shall set `udState` either to `failed` (also setting an appropriate value in `udCause`) or to `disconnected`.

A management terminal with privilege level `operator` or above may disconnect a flow at its source, by setting `usCause` (if required) and `usState` in the same way as for a destination,

¹ Additional subparts are under consideration. See Introduction.

provided its privilege level is also at least as high as the current `usPrivilege` value for the entry. This will disconnect all destinations of the flow.

A management terminal with privilege level `supervisor` or above may disconnect a flow at a switch through which it passes, in the same way as for end equipment, provided its privilege level is also at least as high as the current `udPrivilege` or `usPrivilege` value for the entry. Disconnection by setting `udState` disconnects all destinations reached through the switch. Disconnection by setting `usState` for a port disconnects all destinations reached through that port.

6.4 Maintaining calls

On power down or during a reset operation that causes existing calls to be dropped, a unit shall remove all entries from its `unitDestList` where the value of `ucRemembered` is false.

On power up, if power has been removed for less than an implementation-specific length of time (which is not specified by this standard), a unit shall set the value of `udState` in all remaining entries in its `unitDestList` to `readyToConnect`, otherwise it shall remove all remaining entries from its `unitDestList`.

If a flow is terminated due to normal disconnection, the managed unit shall set its state to `disconnected`. If a flow is terminated for any other reason then, if the value of `ucRemembered` is true, the unit shall set the value of `udState` in the corresponding entry in its `unitDestList` to `readyToConnect`, otherwise it shall set it to `failed`.

After setting the value of `ucState` to `readyToConnect` (for any reason other than a Set request), a unit shall wait for a minimum of $(256 - ucPriority)/10$ s, then attempt to connect the flow, as if the associated entry in its `unitDestList` had just been created.

7 Status broadcasts

7.1 General

A status broadcast shall consist of a sequence of “reports”. Each report shall show the current value of a MIB object, and shall be classified as either an “in-cycle report” or a “change report” for the object.

Each status broadcast stream shall report a defined set of objects, and shall be divided into “cycles”. Each cycle shall contain at least one report for each object in the set as described below.

The set should not be assumed to be static. For instance, it may consist of certain columns in the list of destinations, in which case objects will be added or removed as records are created or destroyed.

Each object in the set shall be in one of the following states:

- a) pending, value has not been reported in the current cycle;
- b) changed, value has changed since it was last reported;
- c) reported, value has been reported in the current cycle.

An object added to the set is initially in “changed” state. Whenever the value of an object in the set changes, it enters the “changed” state, except that the transition to “changed” state may be delayed until a defined time has elapsed since the previous occasion on which it entered the “changed” state. At the start of a cycle, all objects enter the “pending” state.

Whenever there is an opportunity to transmit a report, if there is an object in “changed” state a change report is transmitted for it (or, if there is more than one such object, for one of them), otherwise, if there is an object in “pending” state, an in-cycle report is transmitted for it (or, if there is more than one such object, for one of them), otherwise nothing is transmitted. When a report is transmitted for an object, it enters the “reported” state.

A change report shows the object's current value. If an object changes several times in quick succession the intermediate values will not necessarily be reported. Implementers should ensure that an object will be in “changed” state if its value is different from that in the most recent report, and also if it has been different at some time since that report.

For objects that may change repeatedly, a limit may be placed on the frequency of change reports in order to allow reports of other objects' values to be transmitted. The delay may be specified in relation to other reports, e.g. an additional state may be defined such that there cannot be more than one change report for an object between any two adjacent in-cycle reports, except when there are no objects in “pending” state.

A new cycle shall begin when all objects in the set are in “reported” state and a specified time has elapsed since the start of the previous cycle.

NOTE If a cycle takes less than the allotted time, there will be a period during which nothing is transmitted (except change reports, if any changes occur). If it takes more than the allotted time, all subsequent cycles will be delayed.

7.2 Coding and encapsulation of reports

The coding of each report shall consist of

- a header octet,
- the object's identifier (OID),
- the object's value,
- and a checksum.

Each report should be transmitted in a separate data unit. If a report does not fit in a single data unit, it shall be divided into fragments, with each fragment, except the first, having a header octet (coded for a “continuation fragment”) prepended to it.

NOTE 1 This allows the stream to use small data units, for instance to limit the bandwidth requested if the minimum package rate (see 5.4.2.8 and 5.4.3.12) is 8 kHz.

Each header octet shall be encoded as follows. The most significant two bits shall be coded as

- 00 in-cycle report, not the last in a cycle,
- 01 last in-cycle report in a cycle,
- 10 change report,
- 11 continuation fragment,

and the least significant 6 bit shall contain a sequence number which shall be one more (modulo 64) than in the previous data unit.

NOTE 2 The sequence number allows the recipient to know whether any data units have been lost. The “length” fields in the BER encoding show whether a continuation is expected. In-cycle reports allow updating of the values of objects for which change reports have been lost.

The checksum shall be two octets containing a nonzero value calculated as follows. If there are an even number of octets in the coding of the report, each pair of octets shall be interpreted as a 16-bit unsigned integer, with the most significant half in the first octet of the pair, and the checksum shall be such that the sum of all the 16-bit integers is a multiple of

65 535. If there are an odd number of octets, the checksum shall have the value it would have had if an additional octet containing the value zero had been inserted immediately before the checksum field.

This calculation is the same as for the checksums used in TCP and UDP.

The header octet on the first data unit of a fragmented report (including its sequence number) is considered to be part of the report and is included in the checksum. The header octets on the continuation fragments are not. Thus, the checksum for each report should be calculated when it is ready to be transmitted, but before fragmentation.

If transmission of each report in a separate data unit would be inefficient, multiple reports may be packed into a single data unit. When this encapsulation is used, each report shall have its own header and checksum.

NOTE 3 It follows from the definition of the sequence number that all reports in a data unit will have the same sequence number.

7.3 Standard report groups

7.3.1 General

The following subclauses list “standard” status reports. Units may also implement other groups of objects to be reported, or allow a caller to specify a bespoke set of objects. When sending reports to a large number of destinations, it is clearly better to be able to multicast a standard report than to have to unicast multiple bespoke reports.

7.3.2 List of sources

The “list of sources” report shall be supported by all units that implement `unitSourceList`. It should consist of `usState` in each record.

NOTE This broadcast simply notifies when new connections appear, and when a connection changes state. A recipient can use a Get request to retrieve other information about the connection. It would not be efficient to include in the broadcast objects that may not have meaningful values. A flow can have many destinations, so reporting the destination would not be viable.

7.3.3 List of destinations

The “list of destinations” report shall be supported by all units that implement `unitDestList`. It should consist of `udState`, `udSourceAddrType`, and `udSourceAddress` in each record.

Annex A (informative)

Machine-readable block definitions

This annex provides a machine-readable version of the block definitions which is intended to be interpretable by standard MIB browsing software tools. It does not express all the requirements of the standard, for instance where access to an object is restricted at certain privilege levels. If there is any inconsistency between this annex and 5.4, the information in 5.4 takes precedence.

The format used to describe the MIB objects conforms to IETF STD 58 (SMIv2).

```
IEC62379-5-1-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        BlockId,          Utf8String,          CardinalNumber,          IndexNumber,
        PrivilegeLevel

        FROM IEC62379-1-MIB
        general, NetPortState, ConnectionState, CauseCode, Importance,
        Priority, SourceBlockId, DestBlockId
        FROM IEC62379-5-TC-MIB
        OBJECT-GROUP, MODULE-COMPLIANCE
        FROM SNMPv2-CONF
        OBJECT-TYPE, MODULE-IDENTITY
        FROM SNMPv2-SMI
        TruthValue, TAddress, TDomain, DateAndTime
        FROM SNMPv2-TC;

    -- 1.0.62379.5.1.1
    networkMIB MODULE-IDENTITY
        LAST-UPDATED "201212111500Z"          -- December 11, 2012 at
15:00 GMT
        ORGANIZATION
            "IEC PT62379"
        CONTACT-INFO
            "Not specified"
        DESCRIPTION
            "The MIB module for managing network functions in
IEC 62379
            compliant equipment."
        REVISION "201212111500Z"          -- December 11, 2012 at 15:00
GMT
        DESCRIPTION
            "The MIB module defined in IEC 62379-5-1."
        ::= { general 1 }
    -- Although nodes may have the same name in different
    -- modules, it is recommended that this is avoided
    -- where possible. This "general" node may have to be
    -- renamed as it also exists in IEC 62379-1.

    --
    -- Node definitions
    --

    -- Object identifier values for module compliance statements
    -- 1.0.62379.5.1.1.0
    networkMIBCompliance OBJECT IDENTIFIER ::= { networkMIB 0 }
```

```

-- Compliance statements
-- this module
    -- 1.0.62379.5.1.1.0.1
    networkMIBComplianceV1 MODULE-COMPLIANCE
        STATUS current
        DESCRIPTION
            "The compliance statement for entities that conform to
            IEC 62379-5-1 (2014)."
```

MODULE -- this module

```

        GROUP networkPortsGroup
            DESCRIPTION
                "Mandatory for equipment that contains one
or more network
                ports."
        GROUP callSourcesGroup
            DESCRIPTION
                "Mandatory for equipment that contains one
or more call
                sources."
        GROUP callDestinationsGroup
            DESCRIPTION
                "Mandatory for equipment that contains one
or more
                call destinations."
        ::= { networkMIBCompliance 1 }

-- 1.0.62379.5.1.1.1
networkPorts OBJECT IDENTIFIER ::= { networkMIB 1 }

-- 1.0.62379.5.1.1.1.1
netPortTable OBJECT-TYPE
    SYNTAX SEQUENCE OF NetPortEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.1."
    ::= { networkPorts 1 }

-- 1.0.62379.5.1.1.1.1.1
netPortEntry OBJECT-TYPE
    SYNTAX NetPortEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.1."
    INDEX { nPortBlockId }
    ::= { netPortTable 1 }

NetPortEntry ::=
    SEQUENCE {
        nPortBlockId
            BlockId,
        nPortName
            Utf8String,
        nPortState
            NetPortState,
        nPortAddressType
            TDomain,
        nPortAddress
            TAddress,
        nPortPAddrType
            TDomain,
        nPortPartnerAddr
            TAddress,
        nPortBarred
            TruthValue
    }

```

```

    }

-- 1.0.62379.5.1.1.1.1.1.1
nPortBlockId OBJECT-TYPE
    SYNTAX BlockId
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 1 }

-- 1.0.62379.5.1.1.1.1.1.2
nPortName OBJECT-TYPE
    SYNTAX Utf8String
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 2 }

-- 1.0.62379.5.1.1.1.1.1.3
nPortState OBJECT-TYPE
    SYNTAX NetPortState
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 3 }

-- 1.0.62379.5.1.1.1.1.1.4
nPortAddressType OBJECT-TYPE
    SYNTAX TDomain
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 4 }

-- 1.0.62379.5.1.1.1.1.1.5
nPortAddress OBJECT-TYPE
    SYNTAX TAddress
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 5 }

-- 1.0.62379.5.1.1.1.1.1.6
nPortPAddrType OBJECT-TYPE
    SYNTAX TDomain
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 6 }

-- 1.0.62379.5.1.1.1.1.1.7
nPortPartnerAddr OBJECT-TYPE
    SYNTAX TAddress
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 7 }

```

```

-- 1.0.62379.5.1.1.1.1.8
nPortBarred OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 8 }

-- 1.0.62379.5.1.1.1.2
networkPortsGroup OBJECT-GROUP
    OBJECTS { nPortName, nPortState, nPortAddressType, nPortAddress,
nPortPAddrType,
        nPortPartnerAddr, nPortBarred }
    STATUS current
    DESCRIPTION
        "The group of objects used to control the network port
block."
    ::= { networkPorts 2 }

-- 1.0.62379.5.1.1.2
callSources OBJECT IDENTIFIER ::= { networkMIB 2 }

-- 1.0.62379.5.1.1.2.1
unitSourceListTable OBJECT-TYPE
    SYNTAX SEQUENCE OF UnitSourceListEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { callSources 1 }

-- 1.0.62379.5.1.1.2.1.1
unitSourceListEntry OBJECT-TYPE
    SYNTAX UnitSourceListEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    INDEX { usFlowIdentifier, usBlockId }
    ::= { unitSourceListTable 1 }

UnitSourceListEntry ::=
    SEQUENCE {
        usFlowIdentifier
            OCTET STRING,
        usBlockId
            SourceBlockId,
        usBlockInput
            IndexNumber,
        usPackageSize
            CardinalNumber,
        usPackageRate
            CardinalNumber,
        usPrivilege
            PrivilegeLevel,
        usState
            ConnectionState,
        usCause
            CauseCode,
        usSource
            Utf8String,
        usDestination
    }

```

```

        Utf8String,
        usService
        Utf8String,
        usImportance
        Importance,
        usPriority
        Priority,
        usStartTime
        DateAndTime,
        usEndTime
        DateAndTime,
        usConnectTime
        CardinalNumber,
        usFlowIdStandard
        TruthValue
    }

-- 1.0.62379.5.1.1.2.1.1.1
usFlowIdentifier OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 1 }

-- 1.0.62379.5.1.1.2.1.1.2
usBlockId OBJECT-TYPE
    SYNTAX SourceBlockId
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 2 }

-- 1.0.62379.5.1.1.2.1.1.3
usBlockInput OBJECT-TYPE
    SYNTAX IndexNumber
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 3 }

-- 1.0.62379.5.1.1.2.1.1.4
usPackageSize OBJECT-TYPE
    SYNTAX CardinalNumber
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 4 }

-- 1.0.62379.5.1.1.2.1.1.5
usPackageRate OBJECT-TYPE
    SYNTAX CardinalNumber
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 5 }

-- 1.0.62379.5.1.1.2.1.1.6
usPrivilege OBJECT-TYPE
    SYNTAX PrivilegeLevel

```

```
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "See IEC 62379-5-1, 5.4.2."
::= { unitSourceListEntry 6 }

-- 1.0.62379.5.1.1.2.1.1.7
usState OBJECT-TYPE
    SYNTAX ConnectionState
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 7 }

-- 1.0.62379.5.1.1.2.1.1.8
usCause OBJECT-TYPE
    SYNTAX CauseCode
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 8 }

-- 1.0.62379.5.1.1.2.1.1.9
usSource OBJECT-TYPE
    SYNTAX Utf8String
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 9 }

-- 1.0.62379.5.1.1.2.1.1.10
usDestination OBJECT-TYPE
    SYNTAX Utf8String
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 10 }

-- 1.0.62379.5.1.1.2.1.1.11
usService OBJECT-TYPE
    SYNTAX Utf8String
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 11 }

-- 1.0.62379.5.1.1.2.1.1.12
usImportance OBJECT-TYPE
    SYNTAX Importance
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 12 }

-- 1.0.62379.5.1.1.2.1.1.13
usPriority OBJECT-TYPE
    SYNTAX Priority
```

```

MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "See IEC 62379-5-1, 5.4.2."
::= { unitSourceListEntry 13 }

-- 1.0.62379.5.1.1.2.1.1.14
usStartTime OBJECT-TYPE
    SYNTAX DateAndTime
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 14 }

-- 1.0.62379.5.1.1.2.1.1.15
usEndTime OBJECT-TYPE
    SYNTAX DateAndTime
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 15 }

-- 1.0.62379.5.1.1.2.1.1.16
usConnectTime OBJECT-TYPE
    SYNTAX CardinalNumber
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 16 }

-- 1.0.62379.5.1.1.2.1.1.17
usFlowIdStandard OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.2."
    ::= { unitSourceListEntry 17 }

-- 1.0.62379.5.1.1.2.2
callSourcesGroup OBJECT-GROUP
    OBJECTS {
        usBlockInput,      usPackageSize,      usPackageRate,
        usPrivilege, usState,
        usCause, usSource, usDestination, usService, usImportance,
        usPriority,      usStartTime,      usEndTime,      usConnectTime,
        usFlowIdStandard
    }
    STATUS current
    DESCRIPTION
        "The group of objects used to control the call sources
        block."
    ::= { callSources 2 }

-- 1.0.62379.5.1.1.3
callDestinations OBJECT IDENTIFIER ::= { networkMIB 3 }

-- 1.0.62379.5.1.1.3.1
unitNextFlowId OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write

```

```

STATUS current
DESCRIPTION
    "See IEC 62379-5-1, 5.4.3."
::= { callDestinations 1 }

-- 1.0.62379.5.1.1.3.2
unitNextCallId OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { callDestinations 2 }

-- 1.0.62379.5.1.1.3.3
unitDestListTable OBJECT-TYPE
    SYNTAX SEQUENCE OF UnitDestListEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { callDestinations 3 }

-- 1.0.62379.5.1.1.3.3.1
unitDestListEntry OBJECT-TYPE
    SYNTAX UnitDestListEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    INDEX { udFlowIdentifier }
    ::= { unitDestListTable 1 }

UnitDestListEntry ::=
    SEQUENCE {
        udFlowIdentifier
            OCTET STRING,
        udNetBlockId
            SourceBlockId,
        udNetBlockOutput
            IndexNumber,
        udSourceAddrType
            TDomain,
        udSourceAddress
            TAddress,
        udPackageSize
            CardinalNumber,
        udPackageRate
            CardinalNumber,
        udPrivilege
            PrivilegeLevel,
        udState
            ConnectionState,
        udCause
            CauseCode,
        udSource
            Utf8String,
        udDestination
            Utf8String,
        udService
            Utf8String,
        udImportance
            Importance,
        udPriority
            Priority,
        udStartTime
    }

```

```

        DateAndTime,
        udEndTime
        DateAndTime,
        udConnectTime
        CardinalNumber,
        udConnectCount
        CardinalNumber,
        udRemembered
        TruthValue,
        udDestBlockId
        DestBlockId,
        udDestBlockInput
        IndexNumber,
        udFlowIdStandard
        TruthValue
    }

-- 1.0.62379.5.1.1.3.3.1.1
udFlowIdentifier OBJECT-TYPE
    SYNTAX OCTET STRING
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 1 }

-- 1.0.62379.5.1.1.3.3.1.2
udNetBlockId OBJECT-TYPE
    SYNTAX SourceBlockId
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 2 }

-- 1.0.62379.5.1.1.3.3.1.3
udNetBlockOutput OBJECT-TYPE
    SYNTAX IndexNumber
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 3 }

-- 1.0.62379.5.1.1.3.3.1.4
udSourceAddrType OBJECT-TYPE
    SYNTAX TDomain
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 4 }

-- 1.0.62379.5.1.1.3.3.1.5
udSourceAddress OBJECT-TYPE
    SYNTAX TAddress
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 5 }

-- 1.0.62379.5.1.1.3.3.1.6
udPackageSize OBJECT-TYPE
    SYNTAX CardinalNumber

```

```
MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "See IEC 62379-5-1, 5.4.3."
::= { unitDestListEntry 6 }

-- 1.0.62379.5.1.1.3.3.1.7
udPackageRate OBJECT-TYPE
    SYNTAX CardinalNumber
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 7 }

-- 1.0.62379.5.1.1.3.3.1.8
udPrivilege OBJECT-TYPE
    SYNTAX PrivilegeLevel
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 8 }

-- 1.0.62379.5.1.1.3.3.1.9
udState OBJECT-TYPE
    SYNTAX ConnectionState
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 9 }

-- 1.0.62379.5.1.1.3.3.1.10
udCause OBJECT-TYPE
    SYNTAX CauseCode
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 10 }

-- 1.0.62379.5.1.1.3.3.1.11
udSource OBJECT-TYPE
    SYNTAX Utf8String
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 11 }

-- 1.0.62379.5.1.1.3.3.1.12
udDestination OBJECT-TYPE
    SYNTAX Utf8String
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 12 }

-- 1.0.62379.5.1.1.3.3.1.13
udService OBJECT-TYPE
    SYNTAX Utf8String
```

```

MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "See IEC 62379-5-1, 5.4.3."
::= { unitDestListEntry 13 }

-- 1.0.62379.5.1.1.3.3.1.14
udImportance OBJECT-TYPE
    SYNTAX Importance
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 14 }

-- 1.0.62379.5.1.1.3.3.1.15
udPriority OBJECT-TYPE
    SYNTAX Priority
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 15 }

-- 1.0.62379.5.1.1.3.3.1.16
udStartTime OBJECT-TYPE
    SYNTAX DateAndTime
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 16 }

-- 1.0.62379.5.1.1.3.3.1.17
udEndTime OBJECT-TYPE
    SYNTAX DateAndTime
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 17 }

-- 1.0.62379.5.1.1.3.3.1.18
udConnectTime OBJECT-TYPE
    SYNTAX CardinalNumber
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 18 }

-- 1.0.62379.5.1.1.3.3.1.19
udConnectCount OBJECT-TYPE
    SYNTAX CardinalNumber
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 19 }

-- 1.0.62379.5.1.1.3.3.1.20
udRemembered OBJECT-TYPE
    SYNTAX TruthValue

```

```

MAX-ACCESS read-write
STATUS current
DESCRIPTION
    "See IEC 62379-5-1, 5.4.3."
::= { unitDestListEntry 20 }

-- 1.0.62379.5.1.1.3.3.1.21
udDestBlockId OBJECT-TYPE
    SYNTAX DestBlockId
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 21 }

-- 1.0.62379.5.1.1.3.3.1.22
udDestBlockInput OBJECT-TYPE
    SYNTAX IndexNumber
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 22 }

-- 1.0.62379.5.1.1.3.3.1.23
udFlowIdStandard OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "See IEC 62379-5-1, 5.4.3."
    ::= { unitDestListEntry 23 }

-- 1.0.62379.5.1.1.3.4
callDestinationsGroup OBJECT-GROUP
    OBJECTS { unitNextFlowId, unitNextCallId, udNetBlockId,
udNetBlockOutput, udSourceAddrType,
udSourceAddress, udPackageSize, udPackageRate,
udPrivilege, udState, udCause, udSource, udDestination, udService, udImportance,
udPriority, udStartTime, udEndTime, udConnectTime,
udConnectCount, udRemembered, udDestBlockId, udDestBlockInput,
udFlowIdStandard }
    STATUS current
    DESCRIPTION
        "The group of objects used to control the call
destinations
and call destinations block."
    ::= { callDestinations 4 }

```

END

Annex B (informative)

Machine-readable data formats

This annex provides a machine-readable version of the textual conventions' definitions specific to this standard which is intended to be interpretable by standard MIB browsing software tools. If there is any inconsistency between this annex and 5.2, the information in 5.2 takes precedence.

The format used to describe the textual conventions' definitions conforms to IETF STD 58 (SMIv2).

```
IEC62379-5-TC-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        iec62379
            FROM IEC62379-1-MIB
        MODULE-IDENTITY
            FROM SNMPv2-SMI
        TEXTUAL-CONVENTION
            FROM SNMPv2-TC;

    -- 1.0.62379.5.1.0
    networkTCMIB MODULE-IDENTITY
        LAST-UPDATED "201212111500Z"
        -- December 11, 2012 at
15:00 GMT

        ORGANIZATION
            "IEC PT62379"
        CONTACT-INFO
            "Not specified."
        DESCRIPTION
            "The Textual Conventions MIB module for managing
            network functions in IEC 62379 compliant
            equipment."
        ::= { general 0 }

--
-- Textual conventions
--

NetPortState ::= TEXTUAL-CONVENTION
    STATUS current
    DESCRIPTION
        "The network port state."
    SYNTAX INTEGER
        {
            disabled(1),
            closing(2),
            linkDown(3),
            linkUp(4),
            pointToPoint(5),
            peerGroup(6),
            sharedMedia(7)
        }

-- {
--     disabled      (1),
--     closing       (2),
--     linkDown      (3),
--     linkUp        (4), -- does not know to what linked
--     pointToPoint  (5), -- link is point-to-point
```

```

--      peerGroup          (6), -- e.g. Ethernet hub
--      sharedMedia        (7) -- with master, e.g. 802.11
-- } (disabled..sharedMedia)
--
      PortIdentifier ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
          "An identifier for the port.

          Octet 1 represents the port type (still need to be
defined)
          Octets 2 and 3 represent the port number (high byte in
octet 2)"
        SYNTAX OCTET STRING (SIZE (0..3))

      ConnectionEnd ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
          "The point defining the end of the connection.

          Better description required?"
        SYNTAX INTEGER
          {
            source(1),
            destination(2)
          }
-- {
--   source          (1),
--   destination      (2)
-- } (source..destination)
--
      CauseCode ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
          "Require a description of what CauseCode represents."
        SYNTAX OBJECT IDENTIFIER

      ConnectionState ::= TEXTUAL-CONVENTION
        STATUS current
        DESCRIPTION
          "The connection state of a port.

          Better description required?"
        SYNTAX INTEGER
          {
            readyToConnect(1),
            connectionRequested(2),
            terminating(3),
            active(4),
            failed(5),
            disconnected(6),
            pending(7),
            inactive(8),
            finished(9),
            callProceeding(10),
            receivedOffer(11),
            acceptedOffer(12),
            reservationRequested(13),
            clearing(14)
          }
-- {
--   readyToConnect      (1),
--   connectionRequested (2),
--   terminating         (3),
--   active              (4),
--   failed              (5),
--   disconnected         (6),
--   pending             (7),
--   inactive            (8),
--   finished            (9),

```

```

--      callProceeding      (10),
--      receivedOffer       (11),
--      acceptedOffer       (12),
--      reservationRequested (13),
--      clearing             (14)
--  } (readyToConnect..clearing )
--
--      Importance::= TEXTUAL-CONVENTION
--          STATUS current
--          DESCRIPTION
--              "Network connection importance.
--
--              Better description required?"
--          SYNTAX INTEGER (1..255)
--
--      Priority::= TEXTUAL-CONVENTION
--          STATUS current
--          DESCRIPTION
--              "The priority of this connection.
--
--              Better description required?"
--          SYNTAX INTEGER (1..255)
--
--      SourceBlockId::= TEXTUAL-CONVENTION
--          STATUS current
--          DESCRIPTION
--              "A handle uniquely identifying a source control block
--              within a unit."
--          SYNTAX INTEGER (1..2147483647)
--
--      DestBlockId::= TEXTUAL-CONVENTION
--          STATUS current
--          DESCRIPTION
--              "A handle uniquely identifying a destination control
block
--              within a unit."
--          SYNTAX INTEGER (1..2147483647)
--
--
--  Node definitions
--
--      1.0.62379.5
--      network OBJECT IDENTIFIER::= { iec62379 5 }
--
--      1.0.62379.5.1
--      general OBJECT IDENTIFIER::= { network 1 }
--
--  END

```

Annex C (informative)

Support for future networks

C.1 General

Most current networks use IP (v4 or v6), but there is a growing recognition that this form of packet switching, or, indeed, packet switching in general, has significant shortcomings for much of the traffic carried on the Internet. The service provided by IP networks is "best effort" transmission of individual packets submitted at unpredictable intervals, which is inappropriate for continuous media where transmission of a regular stream of packets with defined latency is required. More details are given in the introduction to IEC 62379-5-2.

C.2 Services provided by the network

This family of standards is mainly concerned with two kinds of service, one suitable for live media and one suitable for management messages. A network may well provide other kinds of service as well, but they are outside the scope of the IEC 62379 series.

The service for live media (including status broadcasts) is intended to be similar to the service provided by a cross-point router. Thus, it is suitable for carrying a stream of data at a constant rate from a source to one or more destinations. Important Quality of Service (QoS) parameters include the maximum and minimum delay between source and destination and the likelihood of parts of the data being lost.

The concept of a "call" or virtual circuit is used. (Formal definitions are given in Clause 3 of IEC 62379-1:2007.) Setting up and tearing down calls corresponds to making and breaking routes in a cross-point router. The normal procedure in router control is to "take" a source from a destination, and the model used in this standard is that the management terminal sends commands to the destination unit as described in C.3.2 below, and the destination unit then asks the network to make the connection.

An interface unit may implement calls in very different ways, depending on the type of network over which the call passes. Network-specific details are given in IEC 62379-5-2 which applies to each type of network. If the network does not support multicasting, the source may send a separate copy of the stream to each destination unit.

If the network offers a connection-oriented service, calls map naturally onto connections and the network can be expected to offer guarantees for the QoS parameters when a call is set up. If the network only offers a connectionless service, calls map onto sessions at a higher layer (for instance, RTP flows which are carried over the connectionless service provided by UDP over IP; note that there is still a "call set-up" process, using a protocol such as SIP), in which case it may only be possible to give an estimate of the QoS parameter values.

The service for management messages may be either connection-oriented or connectionless. In either case, it is assumed to be a "best effort" service which may lose messages or buffer them for an unlimited length of time. On IP networks, the connectionless protocol UDP is used (as specified in RFC 1157), but on other networking technologies there may well be advantages in setting up a call, for instance if authentication can be done at call set-up time rather than separately for each message.

C.3 Network ports, flows, and media streams

C.3.1 Calls and flows

In many cases a call will carry a single media stream, or a pair of similar streams, one in each direction (as in the case of a telephone call). In other cases, it is useful to bundle together several different streams in a single call, for instance programme audio and video (separately) in one direction and talkback in the other.

The term “flow” is used for a single media stream, and a call may therefore carry several different flows. The procedures in this standard connect each flow separately, so that they can be routed differently within the end equipment.

C.3.2 Connectivity model

The physical connection to the network is described by a “block”, as described in 0.3 of IEC 62379-1:2007. The block has an input for each media flow going to the network, and an output for each media flow coming from it. The number of inputs and outputs will usually change dynamically as calls are connected and cleared down. This model is used in switches within the network as well as in end equipment.

Flows of the same type are grouped together independently of the network port through which they pass, for instance the “unit destination list” (see 5.4.3) includes all incoming media streams, and the “unit source list” (see 5.4.2) includes all outgoing media streams.

To “take” the media stream from a remote source, a new entry is first created in the “unit destination list” of the unit that is to receive the stream. Note that at this stage it may not be known on which of its network ports the call will be connected, if it has more than one. When the management terminal has written all the necessary information in this entry, including identifying which input to which block is to receive the incoming media stream, it requests the unit to make the connection on the network. If the connection is successfully made, a new output is created (or an existing, currently unused, output is assigned) on the network port block representing the physical port on which the call was made, and the input that is to receive the media stream (on a block which may represent a media output port or an internal processing function) is connected to it.

If the remote source is already transmitting on the network, and the network supports multicasting, the network will simply copy the existing stream to the new unit. Otherwise, the source unit will receive an incoming call requesting a source which it identifies from information in the call set-up (or INVITE) message. It creates (or assigns) a new input to the relevant network port block and connects it to the source.

C.3.3 Privilege

With each call is associated one of the four privilege levels described in Clause C.5 of IEC 62379-1:2007. In increasing order of privilege they are listener, operator, supervisor, and maintenance. Calls with a given privilege level cannot be affected by management commands issued by a user, or via a management call, with a lower privilege.

C.3.4 Call identity

“Identity” information may be associated with any call, to provide a user with information that may not be apparent from, say, the network addresses of the endpoints. In a broadcast environment this could include an indication of whether it is part of an on-air programme chain, and if so which programme.

Included in this information are “importance” and “reconnection priority”. Importance is associated with a destination, for instance a broadcast transmitter would have the highest importance. A PC on which the programme is being played would have a lower importance.

When reporting the identity of a call which has a large number of destinations, only the higher-importance destinations are reported. Reconnection priority controls the order in which calls are reconnected if an equipment failure causes a large number of calls to fail simultaneously.

More details are given in Clause A.4 of IEC 62379-1:2007.

C.4 Control of routing

For some applications, including radio and TV broadcasting, calls are required to exist for very long periods with very high reliability. IEC 62379 includes facilities to assist that.

Where equipment is duplicated to increase reliability, the control system may request that routes between the two sets of equipment follow different paths in the network, so that no part of the network can become a single point of failure. Destination equipment may receive two copies of the media by different routes.

This allows a call to be "replaced". The replacement route is set up, then the destination equipment switches from using the data from the original route to the replacement, then the original call is torn down. The destination equipment needs to be able to align the two data streams so that there is no discontinuity when switching. This should be carried out using timing information in the data packets.

One use for call replacement is to allow networking equipment to be taken out of service for scheduled maintenance or relocation. This uses another facility, whereby the equipment can be "barred" from accepting new routes. The existing routes through it are then "replaced", and the replacement routes will take a different path. Once all routes through it have been replaced, the equipment can be taken out of service.

C.5 Scheduled calls

If the network supports it, calls to carry live media can be pre-scheduled. If a programme requires a feed from a remote studio or other remote location, the network can be requested in advance to reserve the necessary resources.

This requires all calls with guaranteed QoS to specify the maximum duration of the call, so that the network knows when its resources will have been released. If this duration is exceeded, the network may disconnect the call if required to allow a pre-booked call to be made.

The MIB includes information to support this feature. Where the network requires to know the duration, and it is not specified by the control system, a suitable default (tailored to the application) should be generated by the interface equipment.

Bibliography

IEC 62379 (all parts), *Common control interface for networked digital audio and video products*

RFC 793, *Transmission control protocol – Darpa internet program – Protocol specification*

RFC 768, *User Datagram Protocol*

RFC 1157, *Simple Network Management Protocol (SNMP)*, IETF Standard #15

RFC 3411 to RFC 3418, *Simple Network Management Protocol version 3*, IETF Standard #62

IECNORM.COM : Click to view the full PDF of IEC 62379-5-1:2014

IECNORM.COM : Click to view the full PDF of IEC 62379-5-1:2014

SOMMAIRE

AVANT-PROPOS	46
INTRODUCTION.....	48
1 Domaine d'application	49
2 Références normatives	49
3 Termes, définitions et abréviations	49
3.1 Termes et définitions	49
3.2 Abréviations.....	49
4 Spécifications de service de réseau.....	50
4.1 Service pour support en direct	50
4.2 Services pour messages de gestion.....	50
5 Définitions de la MIB applicables à tous les réseaux.....	50
5.1 Généralités	50
5.2 Définitions de type	50
5.3 Définitions des types de rangées conceptuelles	51
5.4 Définitions d'un objet de la MIB.....	52
5.4.1 Ports de réseau	52
5.4.2 Liste des sources de support	54
5.4.3 Liste des destinations de support en direct.....	57
6 Appels	61
6.1 Liste des destinations dans les matériels d'extrémité	61
6.2 Connexion d'un flux	62
6.3 Terminaison d'un flux.....	63
6.4 Maintien des appels	63
7 Diffusions d'état.....	64
7.1 Généralités	64
7.2 Codage et encapsulation des rapports	65
7.3 Groupes de rapports normalisés	66
7.3.1 Généralités.....	66
7.3.2 Liste des sources.....	66
7.3.3 Liste des destinations	66
Annexe A (informative) Définitions des blocs lisibles par une machine.....	67
Annexe B (informative) Formats de données lisibles par une machine	79
Annexe C (informative) Prise en charge des réseaux futurs	82
C.1 Généralités	82
C.2 Services fournis par le réseau.....	82
C.3 Ports de réseau, flux et flux de support.....	83
C.3.1 Appels et flux.....	83
C.3.2 Modèle de connectivité	83
C.3.3 Privilège	83
C.3.4 Identité d'appel.....	83
C.4 Commande du routage.....	84
C.5 Appels programmés	84
Bibliographie.....	85
Tableau 1 – Objets gérés pour les ports de réseau	52

Tableau 2 – Objets gérés acheminant la liste des sources	54
Tableau 3 – Objets gérés acheminant la liste des destinations.....	57

IECNORM.COM : Click to view the full PDF of IEC 62379-5-1:2014

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

INTERFACE DE COMMANDE COMMUNE POUR PRODUITS AUDIO ET VIDÉO NUMÉRIQUES CONNECTÉS EN RÉSEAU –

Partie 5-1: Transmission sur des réseaux – Généralités

AVANT-PROPOS

- 1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
- 2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
- 3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
- 4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
- 5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
- 6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
- 7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
- 8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
- 9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 62379-5-1 a été établie par le domaine technique 4: Interfaces du système numérique et protocoles, du comité d'études 100 de l'IEC: Systèmes et équipements audio, vidéo et services de données.

Le texte de cette norme est issu des documents suivants:

CDV	Rapport de vote
100/2107/CDV	100/2304/RVC

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette norme.

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

Le comité a décidé que le contenu de cette publication ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "<http://webstore.iec.ch>" dans les données relatives à la publication recherchée. A cette date, la publication sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

IECNORM.COM : Click to view the full PDF of IEC 62379-5-1:2014

INTRODUCTION

Structure de la famille de normes

L'IEC 62379 définit l'Interface de commande commune qui est un protocole pour gérer des matériels audiovisuels connectés en réseau. Les parties suivantes existent ou sont prévues:

- 1 Généralités
- 2 Audio
- 3 Vidéo
- 4 Données
- 5 Transmission sur des réseaux
- 6 Service de transfert par paquets
- 7 Mesure

L'IEC 62379-1:2007, définit les aspects communs à tous les matériels et contient une introduction à l'Interface de commande commune.

Les IEC 62379-2:2008, IEC 62379-3 (à l'étude) and IEC 62379-4 (à l'étude) définissent la commande de fonctions internes spécifiques à des matériels acheminant des types de support particuliers en direct. L'IEC 62379-4 se réfère à des données à contrainte de temps telles que des commandes de matériels d'automatisation, mais pas à des données en paquets telles que les messages de commande eux-mêmes.

L'IEC 62379-5 définit la commande de transmission de ces supports sur chacune des technologies de réseau. Elle inclut les interfaces de gestion spécifiques ainsi que les éléments de commande spécifiques des réseaux intégrés dans le cadre de commande.

L'IEC 62379-5-1, (la présente norme) définit la gestion des aspects communs à toutes les technologies de réseau.

L'IEC 62379-5-2 définit les protocoles pouvant être utilisés entre les matériels de mise en réseau pour permettre l'établissement d'appels qui sont acheminés par différentes technologies de mise en réseau.

À partir de l'IEC 62379-5-3, on définit la gestion d'aspects particuliers à des technologies de mise en réseau individuelles.

L'IEC 62379-6, définit le transport des messages de commande et d'état ainsi que des données non audiovisuelles sur des transports ne prenant pas en charge l'audio et la vidéo, par exemple les liaisons série RS232, avec (comme à l'IEC 62379-5) une partie secondaire séparée pour chaque technologie.

L'IEC 62379-7 définit des aspects spécifiques à la mesure du service auquel sont soumis les flux audio et vidéo et en particulier aux exigences du Groupe de mesures ECN-IPM de l'UER.

INTERFACE DE COMMANDE COMMUNE POUR PRODUITS AUDIO ET VIDÉO NUMÉRIQUES CONNECTÉS EN RÉSEAU –

Partie 5-1: Transmission sur des réseaux – Généralités

1 Domaine d'application

La présente partie de l'IEC 62379 spécifie des aspects de l'interface de commande commune, qui sont communs à toutes les technologies de réseau, incluant la création et la rupture de sessions et le service fourni par le réseau.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 62379-1:2007, *Interface de commande commune des produits audio et vidéo numériques connectés en réseaux – Partie 1: Généralités*

IEC 62379-5-2, *Common control interface for networked digital audio and video products – Part 5-2: Transmission over networks – Signalling* (disponible en anglais seulement)

3 Termes, définitions et abréviations

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans l'IEC 62379-1 et l'IEC 62379-5-2, ainsi que les suivants s'appliquent.

3.1.1

port de support

source ou destination de données de support dans une unité d'interface

Note 1 à l'article: Soit un port physique (par exemple, un connecteur audio ou vidéo externe sur l'unité), soit un port logique (par exemple, une liaison interne à une autre partie de l'unité).

3.1.2

commutateur

élément de réseau acheminant des données de support et d'autres messages entre des liaisons

3.2 Abréviations

TCP	Protocole de commande de transmission	<i>Transmission Control Protocol</i> ^a
UDP	Protocole datagramme d'utilisateurs	<i>User Datagram Protocol</i> ^b
MIB	Base d'informations de gestion	<i>Management Information Base</i>
^a	See RFC 793.	
^b	See RFC 768.	

4 Spécifications de service de réseau

4.1 Service pour support en direct

Un support en direct (incluant des diffusions d'état) doit être transmis en utilisant un service pour lequel, si le réseau le prend en charge, des niveaux garantis de débit, retard et perte de données doivent être exigés.

4.2 Services pour messages de gestion

Il convient de transmettre les messages de gestion en unités de données sur un flux asynchrone comme spécifié dans l'IEC 62379-5-2. Si un tel service n'est pas disponible, un service de datagramme sans connexion, par exemple UDP, peut être utilisé.

Lorsqu'un service orienté connexion est utilisé, au moins un appel à chaque niveau de privilège doit être accepté par une unité de destination à un instant donné quelconque. Si un plus grand nombre d'appels à un niveau de privilège sont acceptés, ceci ne doit pas empêcher l'acceptation d'au moins un appel à chaque autre niveau de privilège.

5 Définitions de la MIB applicables à tous les réseaux

5.1 Généralités

La structure du MIB doit être comme spécifié dans l'IEC 62379-1.

5.2 Définitions de type

Les types suivants au niveau application doivent être utilisés:

```
NetPortState ::= INTEGER {
    disabled          (1),
    closing           (2),
    linkDown          (3),
    linkUp            (4), -- ne sait pas à quoi il est relié
    pointToPoint      (5), -- liaison point à point
    peerGroup         (6), -- p.e. Hub Ethernet
    sharedMedia       (7)  -- avec maître, p.e. 802.11
} (disabled..sharedMedia)
```

```
PortIdentifier ::= OCTET STRING (SIZE(3))
' octet 1 = port type
' octets 2 and 3 = port number (high byte in octet 2)
```

```
ConnectionEnd ::= INTEGER {
    source          (1),
    destination     (2)
} (source..destination)
```

```
CauseCode ::= OBJECT IDENTIFIER
```

NOTE Parce que des codes peuvent être définis dans d'autres parties de la série IEC 62379-5 ou ailleurs.

```
ConnectionState ::= INTEGER {
    readyToConnect    (1),
    connectionRequested (2),
    terminating       (3),
    active             (4),
    failed             (5),
```

```

    disconnected          (6),
    pending              (7),
    inactive             (8),
    finished             (9),
    callProceeding      (10),
    receivedOffer        (11),
    acceptedOffer        (12),
    reservationRequested (13),
    clearing             (14)
} (readyToConnect..clearing )

Importance ::= INTEGER (1..255)

Priority ::= INTEGER (1..255)

```

5.3 Définitions des types de rangées conceptuelles

Les types suivants sont utilisés pour spécifier la syntaxe des objets gérés par la présente norme, représentant des rangées conceptuelles de tableaux.

```

NetPortEntry ::= SEQUENCE {
    nPortBlockId      BlockId,
    nPortName         Utf8String,
    nPortState        NetPortState,
    nPortAddressType  TDomain,
    nPortAddress       TAddress,
    nPortPAddrType    TDomain,
    nPortPartnerAddr  TAddress,
    nPortBarred       TruthValue
}

UnitSourceEntry ::= SEQUENCE {
    usFlowIdentifier  OCTET STRING,
    usBlockId        SourceBlockId,
    usBlockInput      IndexNumber,
    usPackageSize     CardinalNumber,
    usPrivilege       PrivilegeLevel,
    usState           ConnectionState,
    usCause           CauseCode,
    usSource          Utf8String,
    usDestination     Utf8String,
    usService         Utf8String,
    usImportance      Importance,
    usPriority         Priority,
    usStartTime       DateTime,
    usEndTime         DateTime,
    usConnectTime     CardinalNumber,
    usFlowIdStandard  TruthValue
}

UnitDestEntry ::= SEQUENCE {
    udFlowIdentifier  OCTET STRING,
    udNetBlockId      SourceBlockId,
    udNetBlockOutput  IndexNumber,
    udSourceAddrType  TDomain,

```

```

udSourceAddress  TAddress,
udPackageSize    CardinalNumber,
udPrivilege      PrivilegeLevel,
udState          ConnectionState,
udCause          CauseCode,
udSource         Utf8String,
udDestination    Utf8String,
udService        Utf8String,
udImportance     Importance,
udPriority        Priority,
udStartTime      DateTime,
udEndTime        DateTime,
udConnectTime    CardinalNumber,
udConnectCount   CardinalNumber,
udRemembered     TruthValue,
udDestBlockId    DestBlockId,
udDestBlockInput IndexNumber,
usFlowIdStandard TruthValue
}

```

5.4 Définitions d'un objet de la MIB

5.4.1 Ports de réseau

5.4.1.1 Généralités

Chaque connexion physique à un réseau, dans un commutateur ou des matériels d'extrémité, doit être représentée en utilisant un bloc de ports de réseau. Un bloc de ports de réseau doit avoir une entrée pour chaque flux de support sortant et une sortie pour chaque flux de support entrant.

Le groupe d'objets du Tableau 1 doit être mis en œuvre par tous les commutateurs. Le nœud racine pour ces objets doit être

```
{ iso(1) standard(0) iec62379(62379) network(5) general (1) networkMIB(1) networkPorts(1) }
```

Ce nœud doit être utilisé comme identifiant de type de bloc pour les blocs de ports de réseau.

Tableau 1 – Objets gérés pour les ports de réseau

Identifiant	Syntaxe	Index	Lisible	Inscriptible	Volatile	Etat
netPortTable(1)	SEQUENCE OF NetPortEntry		aucun	aucun	oui	obligatoire
└netPortEntry(1)	NetPortEntry		aucun	aucun	oui	obligatoire
└└nPortBlockId(1)	BlockId	oui	aucun	aucun	non	obligatoire
└└└nPortName(2)	Utf8String		auditeur	superviseur	non	obligatoire
└└└nPortState(3)	NetPortState		auditeur	superviseur	oui	obligatoire
└└└nPortAddressType(4)	TDomain		auditeur	aucun	non	obligatoire
└└└nPortAddress(5)	TAddress		auditeur	aucun	non	obligatoire
└└└nPortPAddrType(6)	TDomain		auditeur	aucun	oui	obligatoire
└└└nPortPartnerAddr(7)	TAddress		auditeur	aucun	oui	obligatoire
└└└└nPortBarred(8)	TruthValue		auditeur	superviseur	non	obligatoire

5.4.1.2 netPortTable

Tableau de descripteurs de ports de réseau pour cette unité. Chaque port de réseau physique sur l'unité doit avoir une entrée correspondante dans ce tableau. Il peut également y avoir des entrées pour des ports de réseau «virtuels».

5.4.1.3 netPortEntry

Entrée dans le tableau de ports de réseau.

5.4.1.4 nPortBlockId

Identifiant de bloc pour ce port. Utilisé comme index lors de l'accès au tableau de ports de réseau.

5.4.1.5 nPortName

Nom assigné à ce port. Il s'agit d'une chaîne textuelle arbitraire assignée par le gestionnaire du système. Il convient que cette assignation persiste après des réinitialisations de l'unité.

Tant qu'un nom n'a pas été assigné, cet objet doit avoir une valeur associée à un marquage visible associé au connecteur physique du port.

Noter qu'il convient que nom d'un port sur un commutateur de réseau dont le connecteur est marqué «2» sur l'enveloppe de l'unité soit par défaut à une valeur telle que «Port 2», «Port 2 Ethernet» ou «Port 2 Face avant». S'il est connecté à une prise de réseau, par exemple dans le studio 6, un superviseur peut alors le renommer «Studio 6».

5.4.1.6 nPortState

Etat courant de la couche liaison de la connexion réseau du port.

Si un terminal de gestion règle cet objet à `closing` ou `linkDown`, l'unité gérée doit rerouter. Ou, si cela n'est pas possible, fermer progressivement tous les appels traversant le port. Dans le cas de `closing`, le port doit entrer dans l'état `disabled`.

Tant que le port est dans l'état `linkDown`, l'unité gérée doit tenter d'établir une connexion de réseau sur le port.

5.4.1.7 nPortAddressType

Type d'adresse réseau utilisée pour `nPortAddress`.

5.4.1.8 nPortAddress

Adresse identifiant le port.

NOTE Il s'agit normalement de l'adresse MAC sur 48 bits de l'interface. On peut utiliser une adresse IP si elle est assignée de manière permanente, mais pas si elle est acquise par l'intermédiaire de DHCP.

5.4.1.9 nPortPartnerAddrType

Type d'adresse réseau utilisée pour `nPortPartnerAddr`.

5.4.1.10 nPortPartnerAddr

Dans l'état `pointToPoint`, adresse de l'unité à laquelle le port est connecté.

Dans l'état `sharedMedia`, adresse de l'unité qui commande le réseau local auquel le port est connecté, par exemple la station de base sans fil ou l'horloge maître.

Dans les autres états, la valeur `nPortPartnerAddr` n'est pas définie par la présente norme.

NOTE 1 Cet objet est destiné à permettre à un terminal de gestion «d'explorer» un réseau pour découvrir sa topologie et les ressources qui sont présentes. L'adresse lui permet d'effectuer une connexion de gestion au partenaire de la liaison dans le cas d'une liaison point à point ou, dans le cas d'un segment de réseau à support partagé, à une unité pouvant être capable de délivrer une liste de toutes les unités sur le segment. Par opposition à `nPortAddress`, il convient que la valeur `nPortPartnerAddr` identifie l'unité plutôt que son interface, de sorte que l'EUI-64 est appropriée.

NOTE 2 Pour certains types de segment de réseau, tel qu'un segment Ethernet utilisant CSMA/CD, il n'y a pas de méthode directe pour énumérer toutes les unités présentes sur le segment.

5.4.1.11 nPortBarred

`false` (valeur par défaut) si l'unité est autorisée à se connecter à une route par l'intermédiaire du port; `true` en cas d'interdiction.

5.4.2 Liste des sources de support

5.4.2.1 Généralités

La «liste des sources» comporte une entrée pour chaque flux synchrone transmis par l'unité gérée.

Le groupe d'objets du Tableau 2 doit être mis en œuvre par tous les matériels d'extrémité pouvant constituer la source d'un flux synchrone et par tous les commutateurs. Le nœud racine pour ces objets doit être

{ iso(1) standard(0) iec62379(62379) network(5) general(1) networkMIB(1) callSources(2) }

NOTE 1 Les appels sont toujours connectés par la destination, de sorte que ce tableau est en lecture seule, hormis la possibilité d'effacer un appel de l'extrémité source. Les appels de gestion ne sont pas inclus dans cette liste.

NOTE 2 On suppose que les appels entrants spécifient une source d'une certaine manière pouvant être au moins partiellement dépendante du réseau et à chaque fois qu'une nouvelle connexion est établie, une entrée apparaît dans ce tableau, et disparaît à nouveau lorsque l'appel est libéré.

Tableau 2 – Objets gérés acheminant la liste des sources

Identifiant	Syntaxe	Indice	Lisible	Inscriptible	Volatil	Syntaxe
unitSourceListTable(1)	SEQUENCE OF UnitSourceEntry		aucun	aucun	oui	obligatoire
UnitSourceEntry(1)	UnitSourceEntry		aucun	aucun	oui	obligatoire
usFlowIdentifier(1)	OCTET STRING	oui	aucun	aucun	oui	obligatoire
usBlockId(2)	SourceBlockId	oui	auditeur	aucun	oui	obligatoire
usBlockInput(3)	IndexNumber		auditeur	aucun	oui	obligatoire
usPackageSize(4)	CardinalNumber		auditeur	aucun	oui	obligatoire
usPackageRate(5)	CardinalNumber		auditeur	aucun	oui	obligatoire
usPrivilege(6)	PrivilegeLevel		auditeur	aucun	oui	obligatoire
usState(7)	ConnectionState		auditeur	voir 6.3	oui	obligatoire
usCause(8)	CauseCode		auditeur	voir 6.3	oui	obligatoire
usSource(9)	Utf8String		auditeur	aucun	oui	obligatoire
usDestination(10)	Utf8String		auditeur	aucun	oui	obligatoire
usService(11)	Utf8String		auditeur	aucun	oui	obligatoire
usImportance(12)	Importance		auditeur	aucun	oui	obligatoire
usPriority(13)	Priority		auditeur	aucun	oui	obligatoire

Identifiant	Syntaxe	Indice	Lisible	Inscriptible	Volatile	Syntaxe
usStartTime(14)	DateandTime		auditeur	aucun	oui	obligatoire
usEndTime(15)	DateandTime		auditeur	aucun	oui	obligatoire
usConnectTime(16)	CardinalNumber		auditeur	aucun	oui	obligatoire
usFlowIdStandard(17)	TruthValue		auditeur	aucun	oui	obligatoire

5.4.2.2 unitSourceListTable

Liste des flux pour lesquels cette unité transmet des données de support.

NOTE Dans le cas de matériels d'extrémité, le tableau donne la liste des flux dont l'unité est la source. Dans le cas d'un commutateur, il donne la liste des informations concernant le début des transmissions vers la ou les destinations, pour tous les flux synchrones traversant l'unité.

5.4.2.3 unitSourceEntry

Entrée dans liste des flux pour lesquels cette unité transmet des données de support.

5.4.2.4 usFlowIdentifier

Chaîne d'octets identifiant le flux. Il convient d'utiliser le format spécifié dans l'IEC 62379-5-2 si celui-ci est disponible; voir 5.4.2.19.

5.4.2.5 usBlockId

Identifiant du bloc de port de réseau pour le port de réseau de l'unité traversé par ce flux.

5.4.2.6 usBlockInput

Numéro d'entrée vers le port de réseau identifié par usBlockId, traversé par ce flux.

NOTE L'entrée associée à cette entrée dans le connectorTable (voir le Tableau 2 de l'IEC 62379-1:2007) montre la sortie de bloc constituant la source du flux de support. L'entrée associée à cette sortie dans le modeTable (voir le Tableau 2 de l'IEC 62379-1:2007) présente le format du support transmis.

5.4.2.7 usPackageSize

Nombre maximal d'octets de charge utile pouvant être transmis dans une seule unité de données sur le flux.

NOTE Celui-ci est la taille négociée pour le flux, qui, lorsqu'elle est multipliée par la valeur usPackageRate, définit la largeur de bande nécessaire.

Ce n'est pas la taille maximale de l'unité de transmission pour les liaisons sur lesquelles le flux sera transmis. Par exemple:

- pour une liaison ATM celui-ci est fixé à la valeur 48, nombre d'octets de charge utile dans une cellule;
- pour un flux RTP, celui-ci est la taille maximale de la charge utile pour les paquets RTP.

5.4.2.8 usPackageRate

Nombre d'unités de données par seconde pouvant être transmises sur le flux.

5.4.2.9 usPrivilege

Niveau de privilège associé à ce flux, qui est le plus haut des niveaux de privilège associés à ses destinations si le réseau fournit cette information, supervisor par ailleurs.

5.4.2.10 **usState**

Etat courant de ce flux. L'état `callProceeding` doit indiquer qu'une demande de connexion entrante a été acceptée et que la confirmation du demandeur est en attente.

5.4.2.11 **usCause**

Cet objet doit être initialisé à une valeur de longueur nulle, qui doit indiquer «annulation d'un appel normal», et doit être fixé par l'unité gérée lorsqu'elle modifie `usState` à `failed` ou `disconnected`. Voir aussi 6.3.

5.4.2.12 **usSource**

Nom de la source pour ce flux.

NOTE Dans le cas de matériels d'extrémité, le nom de la source est spécifié en tant que partie de la définition de l'un des blocs traversé par le signal sur son chemin vers le port de réseau. Dans le cas d'un commutateur, celui-ci est hérité de l'objet `udSource` du flux (voir 5.4.2.16).

5.4.2.13 **usDestination**

Nom de destination pour la destination la plus importante du flux atteinte par l'intermédiaire de ce port de réseau.

5.4.2.14 **usService**

Nom de service pour la destination la plus importante du flux atteinte par l'intermédiaire de ce port de réseau.

5.4.2.15 **usImportance**

Importance de la destination la plus importante du flux atteinte par l'intermédiaire de ce port de réseau.

5.4.2.16 **usPriority**

Priorité de la partie du flux connectée par l'intermédiaire de ce port de réseau.

5.4.2.17 **usStartTime**

Heure à laquelle il est prévu que l'état passe de `pending` à `reservationRequested` ou `active`.

5.4.2.18 **usEndTime**

Heure après laquelle le réseau peut libérer les ressources réservées pour le flux sur ce port.

5.4.2.19 **usConnectTime**

Nombre de secondes pendant lesquelles ce flux a été `active` sur ce réseau ou valeur `CardinalNumber` maximale, en considérant la plus petite valeur; zéro si l'état n'est pas `active`.

5.4.2.20 **usFlowIdStandard**

`true` si `usFlowIdentifier` est dans le format spécifié dans l'IEC 62379-5-2, `false` dans le cas contraire.

Noter que si le format spécifié dans l'IEC 62379-5-2 est utilisé, un flux a le même identifiant dans chaque matériel qu'il traverse. Ainsi, les enregistrements dans lesquels

`usFlowIdStandard` ou (dans la liste de destinations, voir 5.4.3.28) `udFlowIdStandard` est à `true`, même dans des unités gérées différemment, se réfèrent au même flux si, et seulement si, ils ont la même valeur de `usFlowIdentifier` ou de `udFlowIdentifier`.

Noter que si `usFlowIdStandard` est à `false`, il convient que le terminal de gestion considère l'identifiant de flux simplement comme une valeur choisie par l'unité gérée qui, lorsqu'elle est combinée avec `usBlockId`, forme un index de cette table. Ainsi, il convient que le terminal géré ne prévoit aucune relation entre les flux sur des ports différents dans `unitSourceListTable` ou entre des flux dans `unitSourceListTable` et dans `unitDestListTable` (voir 5.4.3), en se fondant sur une comparaison de leurs identifiants; il convient que les relations soient établies par l'intermédiaire du `connectorTable` (voir le Tableau 2 de l'IEC 62379-1:2007).

5.4.3 Liste des destinations de support en direct

5.4.3.1 Généralités

La «liste des destinations» comporte une entrée pour chaque flux synchrone reçu par l'unité gérée.

Le groupe d'objets du Tableau 3 doit être mis en œuvre par tous les matériels d'extrémité pouvant constituer la destination de flux synchrones et par tous les commutateurs. Le nœud racine pour ces objets doit être

```
{iso(1) standard(0) iec62379(62379) network(5) general(1) networkMIB(1) callDestinations(3)}
```

Tableau 3 – Objets gérés acheminant la liste des destinations

Identifiant	Syntaxe	Index	Lisible	Inscriptible	Volatil ^a	Syntaxe
<code>unitNextFlowId(1)</code>	OCTET STRING		auditeur	aucun	oui	voir ^b
<code>unitNextCallId(2)</code>	OCTET STRING		auditeur	aucun	oui	voir 5.4.3.3
<code>unitDestListTable(3)</code>	SEQUENCE OF UnitDestEntry		aucun	aucun	oui	obligatoire
<code>UnitDestEntry(1)</code>	UnitDestEntry		aucun	aucun	oui	obligatoire
<code>udFlowIdentifier(1)</code>	OCTET STRING	oui	aucun	aucun	peut-être	obligatoire
<code>udNetBlockId(2)</code>	SourceBlockId		auditeur	voir 6.1	oui	obligatoire
<code>udNetBlockOutput(3)</code>	IndexNumber		auditeur	aucun	oui	obligatoire
<code>udSourceAddrType(4)</code>	TDomain		auditeur	voir 6.1	peut-être	obligatoire
<code>udSourceAddress(5)</code>	TAddress		auditeur	voir 6.1	peut-être	obligatoire
<code>udPackageSize(6)</code>	CardinalNumber		auditeur	aucun	peut-être	obligatoire
<code>udPackageRate(7)</code>	CardinalNumber		auditeur	aucun	peut-être	obligatoire
<code>udPrivilege(8)</code>	PrivilegeLevel		auditeur	voir 6.1	peut-être	obligatoire
<code>udState(9)</code>	ConnectionState		auditeur	voir 6.1	oui	obligatoire
<code>udCause(10)</code>	CauseCode		auditeur	voir 6.1	oui	obligatoire
<code>udSource(11)</code>	Utf8String		auditeur	aucun	peut-être	obligatoire
<code>udDestination(12)</code>	Utf8String		auditeur	voir 6.1	peut-être	obligatoire
<code>udService(13)</code>	Utf8String		auditeur	voir 6.1	peut-être	obligatoire
<code>udImportance(14)</code>	Importance		auditeur	voir 6.1	peut-être	obligatoire
<code>udPriority(15)</code>	Priority		auditeur	voir 6.1	peut-être	obligatoire
<code>udStartTime(16)</code>	DateandTime		auditeur	voir 6.1	peut-être	obligatoire
<code>udEndTime(17)</code>	DateandTime		auditeur	voir 6.1	peut-être	obligatoire
<code>udConnectTime(18)</code>	CardinalNumber		auditeur	aucun	peut-être	obligatoire

Identifiant	Syntaxe	Index	Lisible	Inscriptible	Volatil ^a	Syntaxe
udConnectCount(19)	CardinalNumber		auditeur	superviseur	peut-être	facultative
udRemembered(20)	TruthValue		auditeur	voir 6.1	non	voir ^b
udDestBlockId(21)	DestBlockId		auditeur	voir 6.1	peut-être	voir ^b
udDestBlockInput(22)	IndexNumber		auditeur	voir 6.1	peut-être	voir ^b
udFlowIdStandard(23)	TruthValue		auditeur	aucun	peut-être	obligatoire
Les appels de gestion ne sont pas inclus dans cette liste. Les flux synchrones sont toujours connectés par la destination, de telle sorte que dans le cas de matériels d'extrémité, chaque entrée dans cette liste est un flux ayant été initialisée par un terminal de gestion par l'intermédiaire du processus détaillé à l'Article 6.						
^a Lorsque la volatilité est indiquée par «peut-être», l'objet est volatil si, et seulement si, udRemembered est à false.						
^b Les objets portant la mention «voir ^b » sont obligatoires pour les matériels d'extrémité mais il convient de retourner noSuchName en unités ne pouvant pas être la destination d'un flux de support.						

5.4.3.2 unitNextFlowId

Valeur d'identifiant de flux, qui n'est pas sous la forme spécifiée dans l'IEC 62379-5-2, pour laquelle un nouvel enregistrement est créé dans le tableau, comme décrit en 6.1. Des demandes Get consécutives pour cet objet doivent retourner des valeurs différentes. Il convient de choisir les valeurs d'une manière rendant maximal le temps avant réutilisation d'une valeur particulière quelconque. Une demande GetNext pour cet objet ne doit pas créer de nouvel enregistrement. La valeur retournée (s'il y a lieu) n'est pas définie par la présente norme.

Tandis que le numéro de port est un index dans la liste de sources, de sorte que les numéros de flux ont seulement besoin d'être uniques pour chaque port, dans la liste de destinations le numéro de flux est le seul index et il doit donc être unique pour tous les ports de l'unité gérée.

Il convient que les unités gérées qui prennent en charge les identifiants de flux dans le format spécifié dans l'IEC 62379-5-2 allouent de manière interne un identifiant d'appel comme si unitNextCallId avait été demandé. La valeur retournée peut être l'identifiant d'appel ou un identifiant de flux basé sur celui-ci ou une certaine autre valeur se référant à celui-ci. Dans tous les cas, l'enregistrement créé doit avoir udFlowIdStandard mis à false, et la valeur retournée doit être différente de n'importe quel identifiant de flux dans le format spécifié dans l'IEC 62379-5-2 pouvant être créé en utilisant un identifiant d'appel retourné en lisant unitNextCallId.

Pour satisfaire à cette exigence, il suffit que la chaîne d'octets soit d'une longueur différente du format spécifié dans l'IEC 62379-5-2 ou de commencer par l'identifiant d'appel alloué de manière interne.

5.4.3.3 unitNextCallId

Nouvelle valeur d'identifiant d'appel sous la forme spécifiée dans l'IEC 62379-5-2, d'où peuvent être déterminés des identifiant de flux, comme décrit en 6.1. Des demandes Get consécutives pour cet objet doivent retourner des valeurs différentes. Il convient de choisir les valeurs d'une manière rendant maximal le temps avant réutilisation d'une valeur particulière quelconque. La valeur (s'il y a lieu) retournée pour une demande GetNext pour cet objet n'est pas définie par la présente norme.

Cet objet est obligatoire pour des matériels d'extrémité prenant en charge des identifiants de flux dans le format spécifié dans l'IEC 62379-5-2, mais il doit retourner noSuchName en unités ne prenant pas en charge ce format et en unités retournant noSuchName pour unitNextFlowId.

5.4.3.4 unitDestList

Liste des flux synchrones reçus par cette unité.

NOTE Dans le cas de matériels d'extrémité, le tableau donne la liste des flux dont l'unité est la destination. Dans le cas d'un commutateur, il donne la liste des informations concernant la réception de la source, pour tous les flux de support traversant l'unité.

5.4.3.5 unitDestEntry

Entrée dans la liste des flux synchrones reçus par cette unité.

5.4.3.6 udFlowIdentifier

Chaîne d'octets identifiant le flux de manière unique dans l'unité. Voir aussi 5.4.3.28.

Noter que ceci est le seul champ d'index, il doit donc être unique dans l'unité, et non simplement pour chaque port.

5.4.3.7 udNetBlockId

Dans les états `inactive` et `readyToConnect`, l'identifiant du bloc de port du réseau pour le port de réseau de l'unité à travers lequel il convient de connecter ce flux ou `nullBlock` si le port de réseau doit être choisi par l'unité gérée. L'unité gérée peut ignorer cette valeur, même si elle n'est pas `nullBlock`.

Dans les autres états, l'identifiant du bloc de port de réseau pour le port de réseau de l'unité traversée par ce flux, où zéro si le flux n'est pas associé à un port de réseau spécifique.

5.4.3.8 udNetBlockOutput

Numéro de sortie du port de réseau identifié par `udNetBlockId`, traversé par ce flux, ou zéro si `udNetBlockId` est égal à zéro ou si la sortie n'a pas encore été choisie.

NOTE L'entrée associée à cette sortie dans le `modeTable` (voir le Tableau 2 de l'IEC 62379-1:2007) présente le format du support reçu, et `connectorTable` (voir le Tableau 2 de l'IEC 62379-1:2007) montre vers quelles entrées de bloc (s'il y a lieu) le flux de support reçu est routé de manière interne. Dans le cas d'un commutateur, ces blocs seront les ports de réseau sur lesquels le flux est fourni en sortie.

5.4.3.9 udSourceAddrType

Type d'adresse réseau utilisée pour `ucSourceAddress`.

5.4.3.10 udSourceAddress

Adresse réseau de la source.

5.4.3.11 udPackageSize

Nombre maximal d'octets de charge utile pouvant être transmis dans une seule unité de données sur le flux.

NOTE Voir des exemples en 5.4.2.7.

5.4.3.12 udPackageRate

Nombre d'unités de données par seconde pouvant être transmises pour le flux.

5.4.3.13 udPrivilege

Niveau de privilège associé à ce flux.

5.4.3.14 udState

État courant de ce flux.

5.4.3.15 udCause

Cet objet doit être initialisé à une valeur de longueur nulle, qui doit indiquer «annulation d'un appel normal», et doit être fixé par l'unité gérée lorsqu'elle modifie `udState` à `failed` ou `disconnected`. Voir aussi 6.3.

5.4.3.16 udSource

Nom de la source pour ce flux.

NOTE Le terminal de gestion n'est pas inscriptible dans cet objet. Le nom de la source est hérité de la source du flux.

5.4.3.17 udDestination

Nom de la destination pour ce flux.

5.4.3.18 udService

Nom du service pour ce flux.

5.4.3.19 udImportance

Importance de ce flux.

5.4.3.20 udPriority

Priorité de reconnexion pour ce flux.

5.4.3.21 udStartTime

Cet objet doit être initialisé à une chaîne d'octets de longueur nulle. L'unité gérée peut le régler à l'heure à laquelle il convient de connecter le flux, voir 6.2.

5.4.3.22 udEndTime

Heure après laquelle le réseau peut libérer les ressources réservées pour ce flux. Une chaîne d'octets de longueur nulle indique que le flux doit rester connecté indéfiniment.

5.4.3.23 udConnectTime

Nombre de secondes pendant lesquelles ce flux a été `active` ou valeur `CardinalNumber` maximale, en considérant la plus petite valeur, zéro si l'état n'est pas `active`.

5.4.3.24 udConnectCount

Nombre de fois où ce flux a été connecté (comporte à la fois un reroutage et une reconnexion automatique). La connexion d'un flux n'est comptée que lorsqu'il a été `active` pendant 1 min au moins. Le compte peut être réinitialisé en écrivant la valeur zéro. L'écriture d'une quelconque autre valeur doit être rejetée avec le code d'erreur `badValue`.

5.4.3.25 udRemembered

S'il est à `true`, indique qu'il s'agit d'un flux «rappelé» qu'il convient de reconnecter après une déconnexion par le réseau (par exemple, en conséquence de la défaillance d'une liaison) ou une réinitialisation de l'unité gérée.

5.4.3.26 udDestBlockId

Lorsque `udState` est à `readyToConnect` ou `active`, indique l'identifiant du bloc auquel il convient de connecter ce flux de manière interne ou `nullBlock` si aucune connexion interne ne doit être réalisée. La valeur dans les autres états n'est pas définie par la présente norme.

Noter qu'il existe deux façons de connecter une entrée d'un bloc à une source externe. L'entrée peut être identifiée par `udDestBlockId` et `udDestBlockInput`, auquel cas, l'unité gérée effectue la connexion interne ou ces objets peuvent être laissés à zéro et la connexion interne est effectuée de manière explicite après que l'unité gérée a mis `udNetBlockOutput` à une valeur non nulle. Le premier cas est un processus plus simple pour le terminal de gestion. Le deuxième permet au terminal de gestion de contrôler le format reçu avant d'effectuer la connexion interne et, si le bloc de destination comporte déjà une connexion, d'attendre que les unités de données soient reçues sur la nouvelle connexion avant d'effectuer la commutation.

5.4.3.27 udDestBlockInput

Numéro d'entrée du bloc identifié par `udDestBlockId`, auquel il convient de connecter ce flux de manière interne.

La valeur de cet objet peut être ignorée par l'unité gérée lors de la connexion d'un flux si `udDestBlockId` est à `nullBlock`, et aussi, si le bloc identifié par `udDestBlockId` ne comporte qu'une entrée.

5.4.3.28 udFlowIdStandard

`true` si `udFlowIdentifier` est dans le format spécifié dans l'IEC 62379-5-2, `false` dans le cas contraire.

Noter que

- si le format spécifié dans l'IEC 62379-5-2 est utilisé, un flux a le même identifiant dans chaque matériel qu'il traverse. Ainsi, les enregistrements dans lesquels `usFlowIdStandard` ou (dans la liste de sources, voir 5.4.2.19) `udFlowIdStandard` est à `true`, même dans des unités gérées différemment se réfèrent au même flux si, et seulement si, ils ont la même valeur de `usFlowIdentifier` ou de `udFlowIdentifier`;
- si `udFlowIdStandard` est à `false`, il convient que le terminal de gestion considère l'identifiant de flux simplement comme une valeur choisie par l'unité gérée qui forme un index de cette table. Ainsi, il convient que le terminal géré ne prévoit aucune relation entre les flux sur des ports différents dans ce tableau ou entre des flux dans ce tableau et dans le Tableau 2 spécifié en 5.4.2, en se fondant sur une comparaison de leurs identifiants. Il convient que des relations soient établies par l'intermédiaire du `connectorTable` (voir le Tableau 2 de l'IEC 62379-1:2007).

6 Appels

6.1 Liste des destinations dans les matériels d'extrémité

Lorsqu'une unité envoie une réponse `GetResponse` incluant une valeur `unitNextFlowId` en réponse à une demande `Get`, elle doit créer une «nouvelle» entrée dans sa `unitDestList` pour cette valeur avec `udFlowIdStandard` mis à `false`.

Dans ce paragraphe, une «nouvelle» entrée est une entrée dans laquelle tous les objets numériques qui ne sont pas spécifiés par ailleurs sont à zéro et tous les objets de chaînes d'octets et d'identifiants d'objets ont une longueur nulle, avec `udPrivilege` mis au privilège de l'expéditeur de la demande `Get`, `udState` inactive, et `udRemembered` `false`.

Lorsqu'une unité reçoit une demande Set pour une colonne dans une entrée dans sa `unitDestList` qui n'existe pas mais où l'index et un identifiant de flux valable dans la forme spécifiée dans l'IEC 62379-5-2, et comporte un identifiant d'appel ayant été précédemment retourné comme réponse à une demande Get pour `unitNextCallId`, elle doit créer une «nouvelle» entrée dans sa `unitDestList` pour cet index avec `udFlowIdStandard` mis à `true`.

Un terminal de gestion ne doit pas envoyer de demande Set pouvant créer une entrée comme décrit à l'alinéa précédent sauf s'il a déjà été établi que l'entrée existe déjà ou que l'index et un identifiant de flux dans la forme spécifiée dans l'IEC 62379-5-2 dans laquelle la valeur de l'identifiant d'appel est une valeur ayant été reçue précédemment en réponse à une demande Get pour `unitNextCallId`.

Noter que si l'unité gérée reçoit une demande Set avec un index pouvant être un identifiant de flux selon l'IEC 62379-5-2 incluant un identifiant d'appel n'ayant pas été retourné précédemment comme réponse à une demande Get pour `unitNextCallId`, il convient qu'elle réponde `noSuchName`. Toutefois, elle peut également créer une nouvelle entrée sans contrôler l'identifiant d'appel.

L'unité gérée doit supprimer une entrée lorsque son `udState` est mis à `finished`, et peut supprimer une entrée pour laquelle `udRemembered` est à `false` et `udState` est à `failed` ou `disconnected` ou `inactive` après une temporisation devant durer au moins 5 min.

Une unité d'extrémité doit refuser une demande Set pour n'importe lequel des objets dont le privilège «inscriptible» figure dans le Tableau 3 comme «voir 6.1», sauf si la demande possède un privilège au moins aussi haut que la valeur `udPrivilege` courante pour l'entrée. Dans le cas des objets autres que `udState`, `udDestination`, `udService`, `udImportance`, et `udPriority`, la demande doit également être refusée si `udState` courant n'est pas `inactive`. Dans le cas de `udState`, la demande doit également être refusée sauf si elle modifie la valeur de `inactive` à `readyToConnect`, ou de n'importe quelle valeur à `terminating`, ou de `failed` à `inactive` ou `readyToConnect`, ou de `failed` ou `disconnected` ou `inactive` à `finished`.

NOTE 1 Un opérateur par exemple peut interférer avec des appels établis par d'autres opérateurs et par des auditeurs mais pas avec ceux qui sont établis par des superviseurs. On suppose qu'un opérateur dans une zone n'a pas accès aux matériels des autres zones et en conséquence n'est pas capable d'interférer avec celui-ci.

Dans le cas d'une unité ne pouvant pas être la destination d'un flux synchrone (comme un commutateur, sauf s'il comporte également des ports de support), les objets dont le privilège «inscriptible» figure dans le Tableau 3 comme «voir 6.1» doivent être en lecture seule sauf comme spécifié en 6.3.

NOTE 2 Ceci signifie que les flux ne peuvent être connectés que depuis la destination, et non depuis un point intermédiaire de la route. Ceci signifie également que les flux peuvent uniquement être «rappelés» à une destination.

6.2 Connexion d'un flux

Un terminal de gestion demandant une connexion d'un flux doit d'abord acquérir un identifiant pour le flux, soit en lisant l'objet `unitNextFlowId` de l'unité de destination, soit en construisant un à partir d'un identifiant d'appel délivré à celui-ci par l'unité de destination. Il doit ensuite mettre les objets dans `unitDestEntry` pour cette identifiant à des valeurs appropriées au flux.

Dans la dernière de ces demandes Set (s'il y en a plusieurs), le terminal de gestion doit mettre `udState` à `readyToConnect`.

Lorsque `udState` est mis à `readyToConnect`, si `udStartTime` est une chaîne d'octets de longueur nulle, l'unité gérée doit tenter de se connecter au flux. Au cours du processus de connexion, `udState` peut prendre d'autres valeurs telles que `callProceeding`, en fonction

des modes opératoires de signalisation utilisés. Ces valeurs peuvent être spécifiées en outre dans l'IEC 62379-5-2 ou une autre partie de l'IEC 62379-5¹ s'appliquant au réseau utilisé pour l'appel.

Si `udFlowIdStandard` est à `true`, il convient d'utiliser les modes opératoires spécifiés dans l'IEC 62379-5-2; la demande initiale sera `AddFlow` si la route est déjà connectée, `FindRoute` dans le cas contraire. Sinon, il convient d'utiliser des modes opératoires spécifiques à la technologie du réseau. Si le flux est connecté avec succès, l'unité gérée doit mettre la valeur de `udState` à `active`. Si la demande est refusée, l'unité doit mettre la valeur de `udState` à `failed` et la valeur de `udCause` au code de cause retourné par le réseau. L'unité peut alors mettre la valeur de `udState` à `inactive` et modifier ensuite l'un quelconque des autres objets dans `unitDestEntry` comme approprié pour retenter la connexion (par exemple, pour essayer une autre adresse). Elle peut également modifier directement `udState` à `readyToConnect`, pour effectuer une nouvelle tentative avec les mêmes paramètres. Si la cause est indiquée comme étant temporaire et que `udState` est toujours à `failed` après un minimum de $(256 - udPriority)/10$ s, l'unité gérée doit alors retenter la connexion.

Si `udStartTime` est un objet `DateTime` valable lorsque `udState` est mis à `readyToConnect`, et si le réseau prend en charge une préréservation, l'unité gérée doit demander de réserver des ressources permettant au flux d'être connecté à l'heure indiquée. Lorsque ce processus se termine avec succès, il doit mettre `udState` à `pending`. Si le réseau ne prend pas en charge une préréservation, l'unité gérée doit mettre immédiatement `udState` à `pending`, et tenter ensuite de connecter le flux au dernier moment de sorte que si le flux est connecté avec succès la transition à `active` se produira avant l'heure indiquée.

6.3 Terminaison d'un flux

A la suite d'une demande `Set`, lorsque `udState` d'une entrée dans une `unitDestList` de l'unité est mise à `terminating`, l'unité gérée doit demander au réseau de déconnecter le flux correspondant, en fournissant la valeur dans `udCause` comme cause. A la fin de la déconnexion, l'unité doit mettre `udState` à `failed` si une erreur a été signalée par le réseau, sinon à `disconnected`. S'il n'y a pas de flux correspondant (par exemple, si l'état précédent était `inactive`) l'unité doit mettre `udState` soit à `failed` (en fixant également une valeur appropriée dans `udCause`), soit à `disconnected`.

Un terminal de gestion avec le niveau de privilège `operator` ou au-dessus peut déconnecter un flux à sa source, en réglant `usCause` (le cas échéant) et `usState` de la même manière que pour une destination, à condition que son niveau de privilège soit également au moins aussi haut que la valeur `usPrivilege` courante pour l'entrée. Ceci déconnecte toutes les destinations du flux.

Un terminal de gestion avec le niveau de privilège `supervisor` ou au-dessus peut déconnecter un flux au niveau d'un commutateur qu'il traverse, de la même manière que pour des matériels d'extrémité, à condition que son niveau de privilège soit également au moins aussi haut que la valeur `udPrivilege` ou `usPrivilege` courante pour l'entrée. La déconnexion par le réglage de `udState` déconnecte toutes les destinations atteintes par l'intermédiaire du commutateur. La déconnexion par le réglage de `usState` pour un port déconnecte toutes les destinations atteintes par l'intermédiaire de ce port.

6.4 Maintien des appels

A la mise hors tension ou lors d'une opération de réinitialisation, ceci provoque l'abandon de tous les appels existants, une unité doit supprimer toutes les entrées de sa `unitDestList` où la valeur de `ucRemembered` est `false`.

¹ Des parties secondaires supplémentaires sont à l'étude. Voir l'introduction.

A la mise sous tension, si l'alimentation a été coupée pendant une durée inférieure à une durée spécifique à la mise en œuvre (qui n'est pas spécifiée par la présente norme), une unité doit régler la valeur de `udState` dans toutes les entrées restantes dans sa `unitDestList` à `readyToConnect`, sinon, elle doit supprimer toutes les entrées restantes de sa `unitDestList`.

Si un flux est terminé en raison d'une déconnexion normale, l'unité gérée doit fixer son état à `disconnected`. Si un flux est terminé pour n'importe quelle autre raison, alors, si la valeur de `ucRemembered` est `true`, l'unité doit mettre la valeur de `udState` dans l'entrée correspondante dans sa `unitDestList` à `readyToConnect`, sinon, elle doit la mettre à `failed`.

Après réglage de la valeur de `ucState` à `readyToConnect` (pour toute autre raison qu'une demande Set), une unité doit attendre une durée minimale de $(256 - ucPriority)/10$ s, et tenter ensuite de se connecter au flux, comme si l'entrée associée dans sa `unitDestList` venait d'être créée.

7 Diffusions d'état

7.1 Généralités

Une diffusion d'état doit être constituée d'une séquence de «rapports». Chaque rapport doit présenter la valeur courante d'un objet de la MIB et doit être classé soit comme «rapport dans un cycle», soit comme «rapport de modification» pour l'objet.

Chaque flux de diffusion d'état doit rapporter un ensemble d'objets défini et doit être divisé en «cycles». Chaque cycle doit contenir au moins un rapport pour chaque objet de l'ensemble comme décrit ci-dessous.

Il convient de ne pas supposer que l'ensemble est statique. Par exemple, il peut être constitué de certaines colonnes de la liste de destinations, auquel cas des objets seront ajoutés ou enlevés à mesure que des enregistrements sont créés ou détruits.

Chaque objet de l'ensemble doit se trouver dans l'un des états suivants:

- a) «pending» (en attente), la valeur n'a pas été rapportée dans le cycle courant;
- b) «changed» (modifié), la valeur a changé depuis la dernière fois où elle a été rapportée;
- c) «reported» (rapporté), la valeur a été rapportée dans le cycle courant.

Un objet ajouté à l'ensemble est initialement dans l'état «changed». A chaque fois que la valeur d'un objet de l'ensemble change, il entre dans l'état «changed», à l'exception du fait que la transition vers l'état «changed» peut être retardée jusqu'à ce qu'une durée définie se soit écoulée depuis l'occasion précédente à laquelle il est entré dans l'état «changed». Au début d'un cycle, tous les objets entrent dans l'état «pending».

A chaque fois qu'il y a une occasion de transmettre un rapport, s'il existe un objet dans l'état «changed», un rapport de changement est transmis pour cela (où, s'il existe plus qu'un seul objet, pour l'un d'entre eux), sinon, s'il existe un objet dans l'état «pending», un rapport dans le cycle est transmis pour cela (où, s'il existe plus qu'un seul objet, pour l'un d'entre eux), sinon, rien n'est transmis. Lorsqu'un rapport est transmis pour un objet, il entre dans l'état «reported».

Un rapport de changement présente la valeur courante de l'objet. Si un objet change plusieurs fois en succession rapide, les valeurs intermédiaires ne seront pas nécessairement rapportées. Il convient que les concepteurs s'assurent qu'un objet sera dans l'état «changed» si sa valeur est différente de celle du rapport le plus récent est aussi si elle a été différente à un certain moment depuis ce rapport.

Pour les objets pouvant changer de façon répétée, on peut imposer une limite à la fréquence des rapports de changement pour permettre la transmission des rapports des valeurs des autres objets. Le retard peut être spécifié en relation avec les autres rapports, par exemple on peut définir un état supplémentaire tel qu'il ne puisse pas y avoir plus d'un rapport de changement pour un objet entre deux quelconques rapports adjacents dans le cycle, sauf s'il n'y a pas d'objets dans l'état «pending».

Un nouveau cycle doit commencer lorsque tous les objets de l'ensemble sont dans l'état «reported» et qu'un temps spécifié s'est écoulé depuis le début du cycle précédent.

NOTE Si un cycle dure moins longtemps que le temps alloué, il se trouvera une période pendant laquelle rien est transmis (à l'exception des rapports de changement si de quelconques changements se produisent); s'il dure plus longtemps que le temps alloué, tous les cycles suivants seront retardés.

7.2 Codage et encapsulation des rapports

Le codage de chaque rapport doit être constitué:

- d'un octet d'en-tête;
- de l'identifiant de l'objet (OID);
- de la valeur de l'objet;
- et d'une somme de contrôle.

Il convient de transmettre chaque rapport dans une unité de données séparée. Si un rapport ne tient pas dans une seule unité de données, il doit être divisé en fragments, chaque fragment doit être différent à l'exception du premier et comporter un octet d'en-tête (codé en «continuation fragment» (fragment de suite)) ajouté devant celui-ci.

NOTE 1 Ceci permet au flux d'utiliser de petites unités de données, par exemple pour limiter la largeur de bande nécessaire si la vitesse minimale de l'ensemble (voir 5.4.2.8 et 5.4.3.12) est de 8 kHz.

Chaque octet d'en-tête doit être codé comme suit. Les deux bits les plus significatifs doivent être codés comme suit:

- 00 rapport dans le cycle, n'est pas le dernier rapport dans un cycle;
- 01 dernier rapport dans le cycle dans un cycle;
- 10 rapport de changement;
- 11 fragment de suite,

et les 6 bit les moins significatifs doivent contenir un numéro de séquence qui doit être supérieur d'une unité (modulo 64) par rapport à l'unité de données précédente.

NOTE 2 Le numéro de séquence permet au destinataire de savoir si des unités de données ont été perdues. Les champs «length» (longueur) dans le codage de BER indiquent si une suite est attendue. Les rapports dans le cycle permettent la mise à jour des valeurs des objets pour lesquels des rapports de changement ont été perdus.

La somme de contrôle doit être constituée de deux octets contenant une valeur non nulle calculée comme suit. S'il y a un nombre pair d'octets dans le codage du rapport, chaque paire d'octets doit être interprétée comme un entier non signé de 16 bits, la moitié la plus significative se trouvant dans le premier octet de la paire, et la somme de contrôle doit être telle que la somme de l'ensemble des entiers de 16 bits soit un multiple de 65 535. S'il y a un nombre impair d'octets, la somme de contrôle doit avoir la valeur qu'elle aurait si un octet supplémentaire contenant la valeur zéro avait été inséré juste devant le champ de somme de contrôle.

Noter que ce calcul est le même que pour les sommes de contrôle utilisées dans TCP et UDP.

L'octet d'en-tête sur la première unité de données d'un rapport fragmenté (incluant son numéro en séquence) est considéré comme faisant partie du rapport et est inclus dans la

somme de contrôle. Les octets d'en-tête ne le sont pas. Ainsi, il convient de calculer la somme de contrôle pour chaque rapport lorsqu'il est prêt à être transmis, mais avant fragmentation.

Si la transmission de chaque rapport dans une unité de données séparées est inefficace, plusieurs rapports peuvent être mis en paquet dans une seule unité de données. Lorsqu'on utilise cette encapsulation, chaque rapport doit avoir ses propres en-tête et somme de contrôle.

NOTE 3 On déduit de la définition du numéro de séquence que tous les rapports dans une unité de données auront le même numéro de séquence.

7.3 Groupes de rapports normalisés

7.3.1 Généralités

Les paragraphes suivants donnent la liste des rapports d'état «normalisés». Les unités peuvent également mettre en œuvre d'autres groupes d'objets à rapporter ou permettre à un demandeur de spécifier un ensemble d'objets personnalisé. Lorsqu'on envoie des rapports à un grand nombre de destinations, il vaut nettement mieux pouvoir multidiffuser un rapport normalisé que de devoir diffuser séparément plusieurs rapports personnalisés.

7.3.2 Liste des sources

Le rapport «liste des sources» doit être pris en charge par toutes les unités mettant en œuvre `unitSourceList`. Il convient qu'il soit constitué de `usState` dans chaque enregistrement.

NOTE Cette diffusion notifie simplement que de nouvelles connexions apparaissent et qu'une connexion change d'état. Un destinataire peut utiliser une demande `Get` pour récupérer d'autres informations relatives à la connexion. Il ne serait pas efficace d'inclure dans la diffusion des objets n'ayant pas de valeurs significatives. Un flux peut avoir un grand nombre de destinations, le rapport de la destination n'est donc pas intéressant.

7.3.3 Liste des destinations

Le rapport «liste des destinations» doit être pris en charge par toutes les unités mettant en œuvre `unitDestList`. Il convient qu'il soit constitué de `udState`, `udSourceAddrType`, et `udSourceAddress` dans chaque enregistrement.

Annexe A (informative)

Définitions des blocs lisibles par une machine

Cette annexe fournit une version lisible par une machine des définitions des blocs qui sont destinées à pouvoir être interprétées par des outils logiciels d'exploration de MIB normalisés. Elle n'exprime pas toutes les exigences de la norme, par exemple, lorsque l'accès à un objet est limité à certains niveaux de privilège. S'il existe de quelconques incohérences entre cette annexe et le 5.4, les informations de 5.4 ont priorité.

Le format utilisé pour décrire les objets de la MIB est conforme à l'IETF STD 58 (SMV2).

```

IEC62379-5-1-MIB DEFINITIONS ::= BEGIN

    IMPORTS
        BlockId,          Utf8String,          CardinalNumber,          IndexNumber,
        PrivilegeLevel

        FROM IEC62379-1-MIB
        general, NetPortState, ConnectionState, CauseCode, Importance,
        Priority, SourceBlockId, DestBlockId
        FROM IEC62379-5-TC-MIB
        OBJECT-GROUP, MODULE-COMPLIANCE
        FROM SNMPv2-CONF
        OBJECT-TYPE, MODULE-IDENTITY
        FROM SNMPv2-SMI
        TruthValue, TAddress, TDomain, DateAndTime
        FROM SNMPv2-TC;

    -- 1.0.62379.5.1.1
    networkMIB MODULE-IDENTITY
        LAST-UPDATED "201212111500Z"          -- 11 décembre 2012 à
        15 h 00 UTC
        ORGANIZATION
            "IEC PT62379"
        CONTACT-INFO
            "Non spécifié"
        DESCRIPTION
            "Module de MIB pour gérer les fonctions de réseau des
            matériels conformes
            à l'IEC 62379."
        REVISION "201212111500Z"          -- 11 décembre 2012 à 15 h 00
        UTC
        DESCRIPTION
            "Module de MIB défini dans l'IEC 62379-5-1."
        ::= { general 1 }

    -- Bien que des nœuds puissent avoir le même nom dans différents
    -- modules, il est recommandé de l'éviter
    -- si possible. Ce nœud "général" peut devoir être
    -- renommé car il existe également dans l'IEC 62379-1.

    --
    -- Définitions des nœuds
    --

    -- Valeurs d'identifiant d'objet pour les déclarations de conformité de module
    -- 1.0.62379.5.1.1.0
    networkMIBCompliance OBJECT IDENTIFIER ::= { networkMIB 0 }

    -- Déclarations de conformité

```

```
-- ce module
-- 1.0.62379.5.1.1.0.1
networkMIBComplianceV1 MODULE-COMPLIANCE
    STATUS current
    DESCRIPTION
        "Déclaration de conformité pour les entités qui sont
conformes à
        1'IEC 62379-5-1 (2014)."
```

MODULE -- ce module

```
    GROUP networkPortsGroup
    DESCRIPTION
        "Obligatoire pour les matériels contenant
un ou plusieurs ports
        de réseau."
    GROUP callSourcesGroup
    DESCRIPTION
        "Obligatoire pour les matériels contenant
une ou plusieurs sources
        de réseau."
    GROUP callDestinationsGroup
    DESCRIPTION
        "Obligatoire pour les matériels contenant
une ou plusieurs destinations
        d'appel."
    ::= { networkMIBCompliance 1 }
```

-- 1.0.62379.5.1.1.1

```
networkPorts OBJECT IDENTIFIER ::= { networkMIB 1 }
```

-- 1.0.62379.5.1.1.1.1

```
netPortTable OBJECT-TYPE
    SYNTAX SEQUENCE OF NetPortEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.1."
    ::= { networkPorts 1 }
```

-- 1.0.62379.5.1.1.1.1.1

```
netPortEntry OBJECT-TYPE
    SYNTAX NetPortEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.1."
    INDEX { nPortBlockId }
    ::= { netPortTable 1 }
```

NetPortEntry ::=

```
    SEQUENCE {
        nPortBlockId
            BlockId,
        nPortName
            Utf8String,
        nPortState
            NetPortState,
        nPortAddressType
            TDomain,
        nPortAddress
            TAddress,
        nPortPAddrType
            TDomain,
        nPortPartnerAddr
            TAddress,
        nPortBarred
            TruthValue
```

```
    }

-- 1.0.62379.5.1.1.1.1.1.1
nPortBlockId OBJECT-TYPE
    SYNTAX BlockId
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 1 }

-- 1.0.62379.5.1.1.1.1.1.2
nPortName OBJECT-TYPE
    SYNTAX Utf8String
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 2 }

-- 1.0.62379.5.1.1.1.1.1.3
nPortState OBJECT-TYPE
    SYNTAX NetPortState
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 3 }

-- 1.0.62379.5.1.1.1.1.1.4
nPortAddressType OBJECT-TYPE
    SYNTAX TDomain
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 4 }

-- 1.0.62379.5.1.1.1.1.1.5
nPortAddress OBJECT-TYPE
    SYNTAX TAddress
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 5 }

-- 1.0.62379.5.1.1.1.1.1.6
nPortPAddrType OBJECT-TYPE
    SYNTAX TDomain
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 6 }

-- 1.0.62379.5.1.1.1.1.1.7
nPortPartnerAddr OBJECT-TYPE
    SYNTAX TAddress
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 7 }
```

```

-- 1.0.62379.5.1.1.1.1.8
nPortBarred OBJECT-TYPE
    SYNTAX TruthValue
    MAX-ACCESS read-write
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.1."
    ::= { netPortEntry 8 }

-- 1.0.62379.5.1.1.1.2
networkPortsGroup OBJECT-GROUP
    OBJECTS { nPortName, nPortState, nPortAddressType, nPortAddress,
nPortPAddrType,
        nPortPartnerAddr, nPortBarred }
    STATUS current
    DESCRIPTION
        "Groupe d'objets utilisés pour commander le bloc de ports
du réseau."
    ::= { networkPorts 2 }

-- 1.0.62379.5.1.1.2
callSources OBJECT IDENTIFIER ::= { networkMIB 2 }

-- 1.0.62379.5.1.1.2.1
unitSourceListTable OBJECT-TYPE
    SYNTAX SEQUENCE OF UnitSourceListEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.2."
    ::= { callSources 1 }

-- 1.0.62379.5.1.1.2.1.1
unitSourceListEntry OBJECT-TYPE
    SYNTAX UnitSourceListEntry
    MAX-ACCESS not-accessible
    STATUS current
    DESCRIPTION
        "Voir IEC 62379-5-1, 5.4.2."
    INDEX { usFlowIdentifier, usBlockId }
    ::= { unitSourceListTable 1 }

UnitSourceListEntry ::=
    SEQUENCE {
        usFlowIdentifier
            OCTET STRING,
        usBlockId
            SourceBlockId,
        usBlockInput
            IndexNumber,
        usPackageSize
            CardinalNumber,
        usPackageRate
            CardinalNumber,
        usPrivilege
            PrivilegeLevel,
        usState
            ConnectionState,
        usCause
            CauseCode,
        usSource
            Utf8String,
        usDestination
    }

```