IEC 62379-5-2

Edition 1.0 2014-03

INTERNATIONAL STANDARD

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

IEC 62379-5-2:2014-03(en)

IEC Central Office
3, rue de Varembé
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 It also gives information on projects, replaced and withdrawn publications.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.

IEC 62379-5-2

Edition 1.0 2014-03

INTERNATIONAL STANDARD

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

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

PRICE CODE **XA**

ICS 33.160; 35.100

ISBN 978-2-8322-1455-8

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

COMMON CONTROL INTERFACE FOR NETWORKED DIGITAL AUDIO AND VIDEO PRODUCTS –

Part 5-2: Transmission over networks – Signalling

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 62379-5-2 has been prepared by technical area 4: Digital system interfaces and protocols 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
100/2050/CDV	100/2158/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.

A bilingual version of this publication may be issued at a later date.

INTRODUCTION

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 specifies management of aspects which are common to all network technologies. IEC 62379-5-3 onwards specify management of aspects which are particular to individual networking technologies.

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

IEC 62379-6 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 5-2: Transmission over networks – Signalling

1 Scope

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

It also specifies encapsulation of digital media within those calls.

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 60958 (all parts), *Digital audio interface*

IEC 62365:2009, *Digital audio – Digital input-output interfacing – Transmission of digital audio over asynchronous transfer mode (ATM) networks*

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

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

IEC 62379-2:2008, *Common control interface for networked digital audio and video products – Part 2: Audio*

IEC 62379-5-1, *Common control interface for networked digital audio and video products – Part 5-1: Transmission over networks – General* [1]

ITU-T Recommendation Q.850, *Usage of cause and location in the digital subscriber signalling system No. 1 and the signalling system No.7 ISDN used part*

AES53, *AES standard for digital audio – Digital input-output interfacing – Sample-accurate timing in AES47* (Audio Engineering Society, New York, NY, USA)

3 Terms and definitions

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

---

1 To be published.

3.1
adjacent
<of two network elements> such that data can be sent from one to the other without passing through any other network element

3.2
asynchronous flow
flow consisting of data units for which the time of arrival at the destination is less important

3.3
asynchronous service
best-effort point-to-point service carrying asynchronous flows

3.4
call identifier
first 12 octets of a flow identifier

Note 1 to entry: See 4.3.

3.5
connectionless data unit
data unit which is not part of a flow

3.6
data unit
sequence of one or more octets which is conveyed across the network as a single unit

3.7
end equipment
equipment that is connected to the network and produces or consumes data units

3.8
extended unique identifier
EUI-64
64-bit globally unique identifier, the first 24 or 36 bits of which are an organizationally unique identifier (OUI) administered by the Institute of Electrical and Electronics Engineers (IEEE)

Note 1 to entry: The extended unique identifier contains 64 bit.

3.9
flow
sequence of data units

3.10
information element
IE
element of a message, in tag-length-value format

3.11
link
means by which data units are conveyed between adjacent network elements

3.12
network delay
time from submission of a data unit to the network by the sender to its delivery to the recipient

3.13
network element
piece of equipment which takes part in the process of conveying data and conforms to this Standard

EXAMPLES: Switches, gateways, and interfaces, but not equipment internal to legacy networks.

3.14
path
sequence of links

3.15
route
path from source to destination of a flow

3.16
route identifier
first 13 octets of a flow identifier (see 4.3), excluding the least significant bit of the 13th octet

3.17
synchronous flow
flow for which the network delay experienced by data units is required to be within specified limits

Note 1 to entry: The size of the units, and also the rate at which units are transmitted, may be fixed or may be variable with a defined upper limit.

3.18
synchronous service
one-to-many service carrying synchronous flows

4 Identification

4.1 Byte order

All multiple-octet quantities shall be coded with the most significant octet first.

4.2 Unit identification

Each switch or end equipment unit shall have assigned to it a globally unique EUI-64.

NOTE The EUI-64 can be derived from the MAC address of one of the unit's interfaces by concatenating the first three octets of the MAC address, hexadecimal FFFE, and the last three octets of the MAC address.

4.3 Flow identifiers

Each flow shall be identified by a 16-octet value formatted as shown in Figure 1.

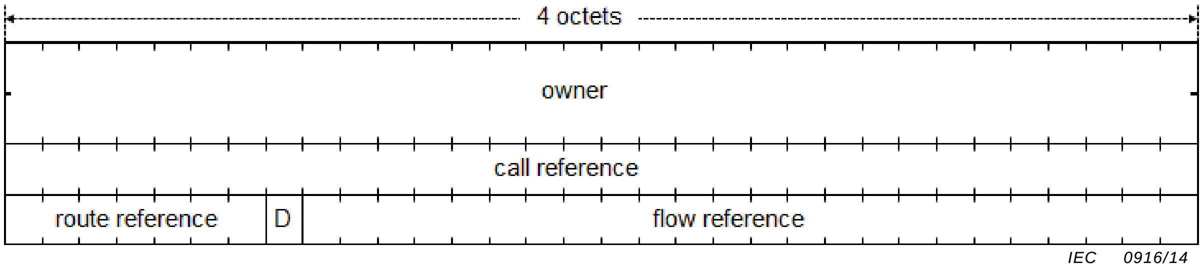


Figure 1 – Structure of flow identifier

The first 8 octets shall contain an EUI-64 identifying the unit which created the flow identifier. This unit is referred to in this standard as the “owner” of the flow identifier.

The next 4 octets shall contain a nonzero reference number, chosen by the owner, identifying the call of which the flow is a part. Call references should be chosen in a way that maximises the time from when a call ceases to exist until its reference is re-used for a new call.

The next octet shall contain, in its most significant seven bits, a nonzero reference number, chosen by the owner and unique within the call, selecting the route followed by the flow; and, in its least significant bit, a “direction” indicator which shall be 0 for a flow in the direction away from the owner and 1 for a flow in the direction towards it.

The remaining 3 octets shall contain a nonzero reference number, chosen by the owner, identifying the flow uniquely within the call. Where a pair of flows in opposite directions carries a two-way protocol such as TCP, the same flow reference value shall be used for each of them.

NOTE 1 The call reference distinguishes between different calls with the same owner. A call may be connected over several different routes (for instance, to give resilience in the event of failure) and the route reference distinguishes between them. The flow reference distinguishes between different flows within a call; for instance, a single call may carry video, programme audio, and talkback, and each of these streams will have its own flow reference.

NOTE 2 The flow reference is orthogonal to the route reference, so flows carrying the same content over different routes have the same flow reference, and flows carrying different content have different flow references even if they do not follow the same route. Different routes may carry different sets of flows, for instance a backup route might only carry the most important flows, or different kinds of flow may have different backup routes.

Zero in the route reference field shall indicate all routes for the call. Except where specified otherwise (e.g. in 5.6.3), zero in the "direction" bit and the flow reference field shall indicate all flows for the route(s), and zero in the flow reference with 1 in the "direction" bit is reserved.

NOTE 3 These identifiers are the same at all points in the system and there is just one set of call reference values for each owner. This contrasts with call references in ITU-T Recommendation Q.2931 where a call has a different reference on every link, and a switch can use the same call reference for different calls as long as they are on different links. See also Clause A.4.

In the case of a unicast call, the owner shall be the caller. In the case of a multicast, the owner shall be the sender or, in the case where for resilience the same content is transmitted by more than one unit, one of the senders. A caller wishing to join a multicast shall use a temporary call identifier (which it owns) until the identifier of the multicast (which may need to be created by the sender) has been discovered.

4.4 Address format

An address shall be represented as an octet string formatted as specified below.

NOTE It is thus a valid `TAddress` value (unless it is more than 255 o long).

The `TDomain` value identifying an address in the form specified in this subclause shall be

{ iso(1) standard(0) iec62379 network(5) signalling(2) address_format(2) }

The first octet of the address shall contain a “type” code; the format and interpretation of the remainder shall depend on the type and on the length of the octet string, as specified in Table 1.

Table 1 – Address type codes

Type	Remainder of octet string
0	second octet contains *n*, third to (*n*+2)th inclusive are an address (which shall not be of type 0) which acts as the locator, remainder is another address (the “local” address, which may be of type 0), which is to be interpreted at the specified location [a]
1	second octet contains *n*, third to (*n*+2)th inclusive are an OID, remainder is a value [b, c, d]
2	second octet contains *n*, third to (*n*+2)th inclusive are an OID, remainder is a value [b, c, e]
3	second octet contains *n*, third to (*n*+2)th inclusive are a TDomain value (which is an OID), remainder is an address within that domain [b, f]
4	IPv4 address if 4 octets, subnet address followed by subnet mask if 8 o
5	EUI-64 identifier
6	IPv6 address
7	URL
8	1 o containing a protocol identifier (as in IP headers, e.g. 6 for TCP and 17 for UDP) followed by a port number for the indicated protocol coded in 2 o as a 16-bit unsigned integer
9	IEC 62379 block identifier
10	service name coded as a UTF8 string
11	NetBIOS unique name
12	NetBIOS group name
13	NSAP address (20 o) or prefix
14	E.164 address
15-255	reserved

[a] In the case of type 0, the locator may be the address of a gateway, or some other identification of a subnetwork (such as an NSAP prefix or an IPv4 subnetwork address). It may also identify end equipment. The locator is not allowed to be a type 0 address so that the sequence of locations will be “flat” rather than nested.

[b] The OID in types 1-3 is coded in the same way as in ASN.1 Basic Encoding Rules (BER), including the compressed form for the first two arcs, but omitting the tag and length. These address types should only be used for address formats that cannot be supported by any of the other types. An IPv4 address, for instance, should use type 4 although it could also be expressed as a type 3 address or, if the target unit’s MIB includes its IP address, as a type 1. An IPv4 address with a port number should be type 0, with a type 4 locator and a type 8 local address.

[c] The value which follows the OID in types 1-2 is coded using ASN.1 Basic Encoding Rules (BER), including the tag and length.

[d] Type 1 identifies a unit in whose MIB the object identified by the OID (such as unitName or unitLocation) has the specified value. In the case of scalar objects, the OID may omit the final zero arc. Units are not required to support all objects in their MIBs that could be used in this context, and switches are not required to remember any but the most obviously useful for the units connected to them. Objects in IEC 62379 should be used in preference to those in other MIBs, for instance unitLocation (1.0.62379.1.1.1.2) rather than sysLocation (1.3.6.1.2.1.1.5).

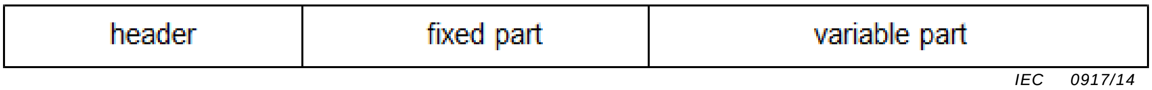
[e] Type 2 identifies a port or other resource within a unit, by searching the unit’s MIB, usually for a columnar object such as aPortName to select an audio port. The OID omits the index arcs. As with type 1, units are not required to support all objects in their MIBs that could be used in this context, so a management terminal which has discovered the port name (e.g. from a status broadcast) should use type 9 instead.

[f] For types 3 onwards, the remainder of the value is an octet string containing the address in the appropriate format. Some codes are only valid in certain contexts, for instance an IEC 62379 block identifier would not be valid on a subnetwork and a NetBIOS name would only be valid within a subnetwork that supported NetBIOS protocols.

5 Message format

5.1 General

All signalling messages shall consist of an octet string with the format shown in Figure 2.



NOTE The header is specified in 5.2 and the variable part in 5.3. The format of the fixed part depends on the message type, and is specified in 5.5.

Figure 2 – Signalling message format

Encapsulation of the octet string for transmission on a link (see 6.1) depends on the technology used for the link, and is not specified in this standard. It should not place a limitation on the maximum length of a message.

5.2 Header

The header of a signalling message shall consist of two octets formatted as shown in Figure 3.

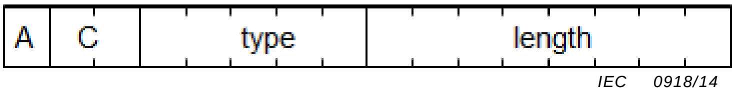


Figure 3 – Signalling message header

The most significant bit of the first octet shall be 1 for an acknowledgement, 0 otherwise. The next two bits shall code the message class as specified in Table 2. The least significant five bits shall code the message type as specified in Table 3.

NOTE For the significance of the “acknowledgement” and “message class” fields, see also the definitions of “original” and “reply”, see 6.1.

The second octet shall contain the number of octets in the fixed part.

Table 2 – Message class

Code	Message class
0	request
1	response
2	confirmation
3	completion

Table 3 – Message type

Code	Message type
0-7	reserved for link management messages [a]
8	FindRoute
9	ClearDown
10	AddFlow
11	NetworkData
12	EndToEndData
13	AsyncSetup
14-15	reserved
16-19	reserved for link-specific resource management messages [a]
20-29	reserved
30-31	extended message types: see 5.5.6

[a] Message types 0-7 and 16-19 may be used for carrying messages that are specific to a particular networking technology, on links that use that technology. Use of such message types should be negotiated when the link is established. Message types 14-15 and 20-29 may be specified in future versions of this standard, or in related standards, including for exchanging network topology information, and should not be used for other purposes.

5.3 Variable part

5.3.1 General

The variable part of a signalling message shall consist of zero or more information elements (IEs), as specified in 5.3.2, which are referred to in this standard as being directly contained or its variable part in the message. The first IE shall begin in the octet immediately following the fixed part of the message, and each subsequent IE shall begin in the octet immediately following the IE that precedes it.

If the encapsulation of the message does not specify the number of octets it occupies, or the number of octets is greater than the length of the message, then the octet immediately following the last IE (or the fixed part if there are no IEs) shall contain zero and any further octets shall be ignored by the recipient of the message.

NOTE A zero in the IE type field is therefore interpreted as "end of message".

5.3.2 Information element format

Each IE shall begin with a 3-octet header with the format shown in Figure 4.

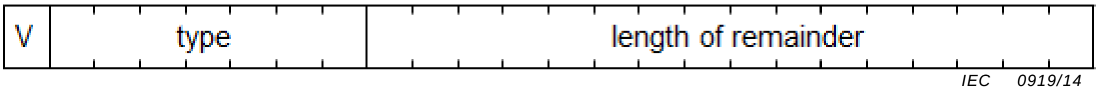


Figure 4 – Information element header

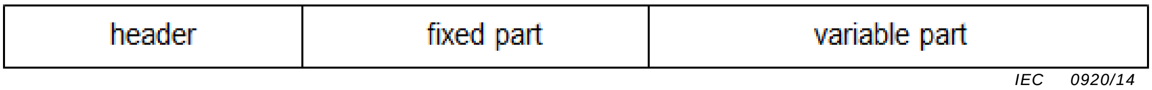
The most significant bit of the first octet shall be 1 if the IE includes a variable part, 0 otherwise.

The least significant seven bits of the first octet shall contain the IE type, coded as specified in 5.6.1.

The next two octets shall contain the number of further octets in the IE.

In the case of an IE with no variable part, the remaining octets shall contain the fixed part.

In the case of an IE with a variable part, the fourth octet shall contain the number of octets in the fixed part. If this number is nonzero, the fixed part shall begin in the fifth octet and the variable part shall begin in the octet immediately following the fixed part, as shown in Figure 5. Otherwise, the variable part shall begin in the fifth octet.



NOTE 1 The format of the fixed part (in either case) depends on the type, and is specified in 5.6.

NOTE 2 If there is a variable part, the fixed part cannot be longer than 255 o, whereas if there is no variable part the format allows the fixed part to be up to 65 535 o.

Figure 5 – Information element with a nonempty fixed part and a variable part

The variable part of an IE (if present) shall consist of zero or more IEs, which are referred to in this standard as being “directly contained” in the IE. The first directly contained or its variable part IE shall begin in the octet immediately following the fixed part, and each subsequent IE shall begin in the octet immediately following the IE that precedes it.

The length signalled in the IE header shall be exactly equal to the sum of the lengths of the fixed part and all the contained IEs.

5.3.3 Order of occurrence of information elements

An IE is “contained” in an IE or message if it is either directly contained in it or directly contained in another IE which is, in turn, contained in it.

An IE is “indirectly contained” in an IE or message if it is contained, but not directly contained, in it.

IEs of the same type that are directly contained within the same variable part shall be grouped together, i.e. there shall not be any directly contained IEs of a different type occurring between any two of them.

NOTE 1 This provision allows the recipient of a message to build a table showing the location of each IE type in the message or in the variable part of an IE, to simplify processing of the message or IE.

NOTE 2 There is no other restriction on the order of occurrence, except where specified in 5.6.23.

5.4 Data formats

Within the fixed part (of a message or an IE), the following conventions as to how various values will be represented shall apply unless the specification explicitly states otherwise.

- OID: coded in the same way as in ASN.1 Basic Encoding Rules (BER), including the compressed form for the first two arcs, but omitting the tag and length.
- RELATIVE OID: coded in the same way as in ASN.1 Basic Encoding Rules (BER), omitting the tag and length.
- integer: most significant octet first; description shows length and whether signed or unsigned.
- octet string: verbatim.
- UTF8 string: octet string; terminating NUL is not required and should not be included.
- address: octet string, as specified in 4.4.

NOTE The length of a value is fixed by the context; often it will be in an unsigned integer immediately preceding the value, or deduced from the length of the fixed part.

5.5 Contents of fixed and variable parts

5.5.1 FindRoute messages

The fixed part of a FindRoute message shall consist of 13 o containing the route identifier for the route, with the least significant bit of the 13th octet coded as zero.

In messages that are not acknowledgements, the variable part shall contain IEs as specified in 6.2 for each message class.

In an acknowledgement message, the variable part shall contain a copy of each McastRoute or InterimOffer IE that was in the original message. There should be no other IEs.

5.5.2 ClearDown messages

The fixed part of a ClearDown request message shall consist of 3 o containing a serial number chosen by the sender. The recipient shall not attribute any significance to the serial number. The variable part shall contain IEs as specified in 6.3 for each message class.

The fixed part of the acknowledgement to a ClearDown request message shall consist of 3 o containing the serial number that was in the original message. The variable part should be empty.

NOTE The only purpose served by the serial number is to show which message is being acknowledged. The recipient cannot, for instance, assume that a gap in serial number values means that a message has been lost.

Other classes (i.e. response, confirmation, and completion) shall not be used with the ClearDown message type.

5.5.3 AddFlow messages

The fixed part of an AddFlow message shall consist of 13 o containing the route identifier for the route, with the least significant bit of the 13th octet coded as zero. The variable part shall contain IEs as specified in 6.4.

5.5.4 NetworkData and EndToEndData messages

The fixed part of a NetworkData or EndToEndData message shall consist of 13 o containing the route identifier for the route, with the least significant bit of the 13th octet coded as zero.

In messages that are not acknowledgements, the variable part shall contain IEs as specified in 6.5.

In an acknowledgement message, the variable part shall contain a copy of each McastRoute or InterimOffer IE that was in the original message. There should be no other IEs.

5.5.5 AsyncSetup messages

The fixed part of an AsyncSetup request message shall consist of 13 o containing the route identifier for the route, with the least significant bit of the 13th octet coded as zero. The variable part shall contain IEs as specified in 6.6.

Other classes (i.e. response, confirmation, and completion) shall not be used with the AsyncSetup message type.

5.5.6 Extended message types

Where the message type field is coded as 31, the fixed part shall begin with an OID, and the format of the remainder of the fixed part, and the semantics and protocols associated with the message, shall be specified by the owner of the OID.

Where the message type field is coded as 30, the fixed part shall begin with a RELATIVE OID rooted at 1.3.6.1.4.1, and the format of the remainder of the fixed part, and the semantics and protocols associated with the message, shall be the same as for a type 31 message with the same OID.

NOTE 1 This allows proprietary and experimental protocols and message types to be carried in standard messages and ensures that they are uniquely identified.

NOTE 2 Type 30 provides a more efficient coding for OIDs descended from the “enterprises” node.

NOTE 3 There is no explicit specification of the length of the OID. The recipient of a message can match each OID it supports against the start of the fixed part of the message. The only way this could be ambiguous is if the owner of an OID has allocated two message types, both descended from that OID, such that one is a subset of the other.

5.6 Information element types

5.6.1 Coding of “type” field

The coding of the least significant seven bits of the first octet of an IE shall be as specified in Table 4.

Table 4 – Information element types

Code	IE type	Refer to
0	reserved for “end of variable part” indication	5.3.1
1, 2	reserved for address registration	Remark
3	CalledAddress	5.6.2
4	FlowDescriptor	5.6.3
5	DataType (format or protocol)	5.6.4
6	StartTime	5.6.5
7	EndTime	5.6.6
8	Importance	5.6.7
9	ServiceName (includes programme name)	5.6.8
10	SourceName	5.6.9
11	DestinationName	5.6.10

Code	IE type	Refer to
12	PrivilegeLevel	5.6.11
13	Password	5.6.12
14	Charge (for call)	5.6.13
15	CallingAddress	5.6.14
16	RouteMetric	5.6.15
17	SyncParams (service parameters)	5.6.16
18	AsyncParams (service parameters)	5.6.17
19	SyncAlloc (link-specific resource allocation)	5.6.18
20	AsyncAlloc (link-specific resource allocation)	5.6.18
21	Delay	5.6.19
22	McastRoute (route identifier of multicast)	5.6.20
23	Cause	5.6.21
24	Route	5.6.22
25	Alternatives	5.6.23
26	Group	5.6.24
27	InterimOffer	5.6.25
28	PathMTU	5.6.26
29	DestCount (number of destinations)	5.6.27
30	DestSelect (destination selection)	5.6.28
31	UserData	5.6.29
32 to 63	reserved	
64 to 71	reserved for link-specific resource management	5.6.18
72 to 125	reserved	
126	ExtendRel (extended IE types with relative OID)	5.6.30
127	ExtendAbs (extended IE types with absolute OID)	5.6.30
REMARK IE types 1 and 2 are intended for use with messages (not specified in this standard) which allow end systems to inform the network of identifiers by which they can be addressed. See also table-footnote [a] to Table 3.		

The coding and semantics of the fixed and (if present) variable parts are specified below.

5.6.2 Called address

The fixed part of a CalledAddress IE shall contain the address of the called party, in the form specified in 4.4. There shall be no variable part.

5.6.3 Flow descriptor

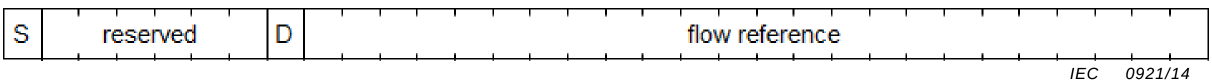


Figure 6 – Fixed part of FlowDescriptor IE

The fixed part of a FlowDescriptor IE shall consist of four octets as illustrated in Figure 6. The most significant bit of the first octet shall be 1 for a synchronous flow and 0 for an asynchronous flow. The least significant bit of the first octet shall be 1 for a flow in the direction towards the owner and 0 for a flow in the direction away from the owner. The remaining bits in the first octet shall be coded as zero by the sender and ignored by the recipient.

The other three octets shall contain a flow reference as specified in 4.3, except that zero shall indicate that the flow reference is unspecified.

NOTE Unlike the case where zero indicates “all flows”, the “direction” flag is still valid when the flow reference is coded as zero.

There may be a variable part which may contain IE types 5-13 and 17-21, either directly or contained (directly or in Group IEs) in one or more Alternatives IEs.

The direction and flow reference shall be combined with a route identifier specified elsewhere in the message, to form the flow identifier for the flow to which the IEs in the variable part, and the flag in the most significant bit of the first octet, apply.

5.6.4 Data format or protocol

The fixed part of a DataType IE shall consist of an OID representing a property of the call, route, or flow to which it applies. There shall be no variable part.

EXAMPLE The FlowDescriptor IE for a flow carrying audio data may carry a DataType IE with an OID descended from an audioSignalFormat (specified in IEC 62379-2) to describe the audio signal, and another DataType IE which describes the encapsulation on the network.

5.6.5 Start time

The fixed part of a StartTime IE shall consist of 9 o coded with a DateTime value (as specified in IEC 62379-1) showing the UTC date and time by which the route or flow is to be connected. There shall be no variable part.

5.6.6 End time

The fixed part of an EndTime IE shall consist of 9 o coded with a DateTime value (as specified in IEC 62379-1) which is the UTC date and time after which the resources reserved for the route or flow may be released. There shall be no variable part.

5.6.7 Call importance

The fixed part of an Importance IE shall consist of 1 o coded with the importance followed by 1 o coded with the reconnection priority. There shall be no variable part.

Importance and reconnection priority shall be as specified in IEC 62379-5-1.

5.6.8 Service (or programme) name

The fixed part of a ServiceName IE shall consist of the service (or programme) name coded as a UTF8 string. There shall be no variable part.

Service name shall be as specified in IEC 62379-5-1.

5.6.9 Source name

The fixed part of a SourceName IE shall consist of the source name coded as a UTF8 string. There shall be no variable part.

Source name shall be as specified in IEC 62379-5-1.

5.6.10 Destination name

The fixed part of a DestinationName IE shall consist of the destination name coded as a UTF8 string. There shall be no variable part.

Destination name shall be as specified in IEC 62379-5-1.

5.6.11 Privilege level

The fixed part of a PrivilegeLevel IE shall consist of 1 o coded with the privilege level. There shall be no variable part.

Privilege level shall be as specified in IEC 62379-1.

5.6.12 Password

The fixed part of a Password IE shall contain a password. There shall be no variable part.

NOTE Format of the password is as required by the called party. Standard formats and security measures are for further study.

5.6.13 Charge for call

The first octet of the fixed part of a Charge IE shall contain flags as detailed below. Beginning at the second octet shall be an OID which identifies the syntax and semantics of the remainder of the IE.

The most significant two bits of the first octet shall be coded as 00 if the IE indicates the maximum the originator of the IE is willing to pay, 01 if the IE indicates a charge which the originator of the IE wishes to levy, or 10 if the IE indicates a charge which the originator of the IE accepts and will pay. The code point 11 is reserved and shall not be used.

NOTE These codings were chosen to match the message class; see 6.2.2, 6.2.3.3, and 6.2.4.3.

The remainder of the first octet is reserved and should be coded as zero by the sender of the message and ignored by the recipient.

EXAMPLE 1 An OID might be defined which indicates a charge which is to be paid by the caller to the caller's ISP. In this case the IE would be originated by the caller (and removed by the ISP) in request and confirmation messages, and originated by the ISP in response messages.

EXAMPLE 2 An OID might be defined which indicates that the charge is to be paid to the called party (as content owner) by a rights management organisation identified by an “extended type” IE in the variable part.

5.6.14 Calling address

The fixed part of a CallingAddress IE shall contain the address of the calling party, in the form specified in 4.4. There may be a variable part containing one or more IEs which provide a measure of the trustworthiness of the address or of the calling party. The coding of these IEs is for further study.

5.6.15 Route metric

The most significant two bits of the first octet of the fixed part of a RouteMetric IE shall contain its status, coded as specified below. The remainder of the first octet, together with the second octet, shall be coded with a 14-bit unsigned integer which is the number of links over which the route passes.

NOTE The coding of the remainder of the IE, including a measure of the proportion of available space required on each link or on the smallest-capacity or most congested link, is for further study.

Inclusion of information beyond the first two octets shall be optional. Equipment designed before the coding of this additional information is defined may continue to send two-octet RouteMetric IEs and should ignore anything beyond the first two octets in incoming IEs.

A status value of 0 shall indicate that the value for the direction in which the message passes is being accumulated, and the value coded is the accumulated value up to and including the link over which the message is transmitted.

A status value of 1 shall indicate the same as status value 0 and additionally that the end-to-end value should be reported back to the originator of the message.

A status value of 2 shall indicate that the value for the direction opposite to that in which the message passes is being accumulated, and the value coded is the accumulated value starting with the link over which the message is transmitted.

A status value of 3 shall indicate that the IE reports the end-to-end value for the opposite direction.

5.6.16 Synchronous service parameters

The fixed part of a SyncParams IE shall consist of eight octets containing two 32-bit unsigned integers. The first integer shall be coded with the maximum number of payload octets in a data unit and the other with the maximum number of data units per second. There shall be no variable part.

NOTE 1 These are the “package size” and “package rate” parameters reported in the list of sources specified in IEC 62379-5-1.

NOTE 2 These parameters refer to the data units that the application will transmit on a flow. The maximum size of data unit supported by the route is reported in the PathMTU IE.

Further details of the traffic contract implied by a SyncParams IE shall be link-specific, and are not specified in this standard.

NOTE 3 Different technologies will allocate resource with different granularity, and may have different rules for traffic shaping and policing. For instance, to transmit stereo audio, a data unit size of 8 o and a rate of 44 100 per second (an average of one every 22,67µs) might be requested. One kind of link may require data units to be sent at 22,67µs intervals, similar to an ATM CBR traffic contract. Another kind may provide an 8 kHz clock and require no more than 6 data units to be sent in each 125 µs clock period.

The value for the maximum number of data units per second shall be the maximum that could be required, including an allowance for inaccuracy of the source frequency.

EXAMPLE Audio sampled at 48 kHz with a relative tolerance of ±0,000 1 % requires 47 999,52 to 48 000,48 data units per second (assuming one data unit per sample) so it is necessary to request 48 001. The network will then round this up to a capacity that it can allocate, which may be different on different links traversed by the route.

5.6.17 Asynchronous service parameters

The fixed part of an AsyncParams IE shall consist of four, eight, twelve, or sixteen octets containing one to four unsigned 32-bit integers. The first integer shall be the mean number of data units to be transmitted per second. The second shall be the mean size of the data units, in octets. The third shall be the burst size specified as follows: assuming a model where data units are written to a buffer pool at the offered rate and transmitted on from that buffer pool at the specified average rate, the burst size shall be the maximum number of data units in the buffer pool. The fourth shall be an estimate of the total number of data units to be transmitted.

In each case zero shall indicate “unspecified” and $FFFFFFFF_{16}$ shall indicate “more than $FFFFFFFE_{16}$”. Where there are less than four integers, the remaining values shall be interpreted to be unspecified.

There shall be no variable part.

NOTE The asynchronous service parameters are intended to give an estimate of the load that the flow will place on the network and on the recipient, but (unlike the synchronous case) not form a basis for QoS guarantees. The parameters for the direction towards the sender of a message indicate the maximum load expected from incoming traffic.

5.6.18 Link-specific resource allocations

The format and interpretation of SyncAlloc and AsyncAlloc IEs shall be specific to the networking technology used on the link over which the message passes.

NOTE These IE types are intended for information related to the low-level allocation of resources on the link, including flow labels (or virtual circuit numbers), port numbers, multicast addresses, etc, to a flow. The format and interpretation are therefore not specified in this standard.

5.6.19 End-to-end delay

A Delay IE shall code the upper and lower bounds to the delay which will be experienced by data units transmitted on the flow if it is (directly or indirectly) inside a FlowDescriptor IE, on any synchronous flows along the route otherwise. In the latter case, the reported figure should cover all possible synchronous flows, not just flows that exist or are being connected at the time the delay is reported.

NOTE Synchronous and asynchronous flows will usually experience different amounts of delay; in particular, asynchronous flows can be expected to experience much larger queuing delays than synchronous flows. Where the delay for a route is requested, the figure for synchronous flows will in most cases be more useful than that for asynchronous flows.

The fixed part of a type 21 IE shall consist of either four or eight octets. There may be a variable part containing one or more IEs which provide a measure of the accuracy of the information in the fixed part. The coding of these IEs is for further study.

The most significant two bits of the first octet of the fixed part shall contain the status, coded in the same way as for a RouteMetric IE (see 5.6.15).

The remainder of the first four octets shall be coded with a 30-bit unsigned integer which is the lower bound to the delay, in sixteenths of a microsecond.

If the fixed part is four octets, this shall indicate that there is no upper bound to the delay.

If the fixed part is eight octets, the second four shall be coded with a 32-bit unsigned integer which is the upper bound to the delay.

5.6.20 Route identifier of multicast

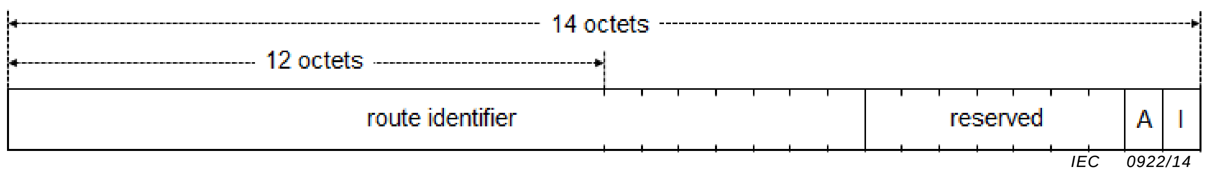


Figure 7 – Fixed part of multicast route identifier IE

The fixed part of a McastRoute IE shall consist of 14 o as illustrated in Figure 7. The first 12 o and the most significant seven bits of the 13th octet shall contain the route identifier assigned by the source. The least significant bit of the 13th octet and the most significant six bits of the 14th octet shall be coded as zero by the sender and ignored by the recipient.

The least significant two bits of the 14th octet shall contain an “openness” value coded as: 00 if the source of the multicast does not wish to be informed when destinations are added or removed; 01 if the source requires to be informed of the total number of destinations; or 11 if addition of a new destination requires the approval of the source. The code point 10 is reserved and shall not be used.

NOTE In Figure 7, “I” indicates that the source is to be informed and “A” that its approval is required.

There shall be no variable part.

5.6.21 Cause

If the fixed part of a Cause IE is one octet in length or is empty, this shall indicate normal call clearing by the remote party and there should be no variable part. The fixed part, if nonempty, should be coded as zero.

If the fixed part is more than one octet long, it shall contain either an OID or a RELATIVE OID indicating the reason for disconnection or refusal to connect, beginning at the second octet. The most significant bit of the first octet shall be coded as 1 if the connection should be retried by another route, 0 otherwise. The least significant two bits of the first octet shall be coded as 00 for an OID, 10 for a RELATIVE OID rooted at 1.0.62379.5.2.4, and 11 for a RELATIVE OID rooted at 1.0.62379.5.2.5 (see Clause 8). The coding 01 is reserved and should not be used. The remainder of the first octet is reserved and should be coded as zero by the sender of the message and ignored by the recipient. The variable part may contain additional information relevant to the condition indicated by the OID.

During call set-up, the first bit should be coded as 1 if the destination could not be reached but might be reachable by another route, as 0 if it was located and refused the call. When clearing down a route, it should be coded as 1 if the cause was a broken link or other equipment failure, 0 if the remote party disconnected.

5.6.22 Route and flow selection

The fixed part of a Route IE shall consist of 13 o containing a route identifier, with the least significant bit of the 13th octet coded as zero. There may be a variable part containing one or more FlowDescriptor IEs.

A Route IE with a variable part shall select the flows indicated by the FlowDescriptor IEs contained in it, but not the route itself, even if it does not carry any other flows.

A Route IE with no variable part shall select the route and all flows carried on it.

5.6.23 Alternatives

The fixed part of an Alternatives IE shall be of zero length. The variable part may contain any number of IEs of one type, which may be any of those with codes 3 to 14 inclusive (see Table 4), or SyncParams or AsyncParams or Group.

An Alternatives IE shall indicate alternatives for the information in the IEs in its variable part, which shall occur in decreasing order of preference.

5.6.24 Group

The fixed part of a Group IE shall be of zero length. The variable part may contain IEs whose type is any of those with codes 3 to 14 inclusive (see Table 4) and/or SyncParams or AsyncParams. The Group IE shall only occur inside an alternatives IE. It shall indicate that the IEs contained in it form a single alternative.

EXAMPLE A call requesting a high-quality video signal but prepared to accept a more heavily-compressed format might include an Alternatives IE containing two group IEs, one for each format, each containing a DataType IE describing the format and a SyncParams IE showing the throughput required to convey it. See A.1.6 for more details.

5.6.25 Interim offer

The fixed part of an InterimOffer IE shall consist of 10 o of which the first eight shall contain an EUI-64 identifying a switch and the remaining two shall contain a serial number, chosen by the switch identified by the first eight octets, which identifies an “offer” uniquely within the context of the switch and the call for which the offer is made. There shall be no variable part.

A FindRoute response message shall be an “interim offer” if it contains at least one InterimOffer IE, a “final offer” if it does not contain any InterimOffer IEs.

5.6.26 Path MTU

The fixed part of a PathMTU IE shall consist of either one or two “packet size records” as specified in this subclause. There shall be no variable part.

If there is just one packet size record, it shall apply to data units on all flows. If there are two packet size records, the first shall apply to synchronous flows and the second to asynchronous flows.

A packet size record shall consist of twelve octets containing three 32-bit unsigned integers as follows:

a) the first integer shall be the maximum number of octets in a data unit, such that a data unit larger than indicated will be either fragmented or discarded;
b) the second shall be the minimum number of octets in a nonempty data unit, such that any data unit with fewer octets will have padding added; and
c) the third shall be the number of overhead octets for a minimum-sized data unit, i.e. the difference between the second number and the number of octets required to transmit a data unit of that size.

In the case of a single link, the values shall reflect the technology used for that link.

EXAMPLE 1 Sending a *n*-octet UDP datagram over Ethernet using tagged packets with Ethertype 0800_{16} requires 12 o inter-packet gap, 8 o preamble, 18 o Ethernet header, 20 o IP header, 8 o UDP header, *n* o data, and 4 o FCS. The first number is 1 500 – (20 + 8) = 1 472, the second is 64 – (18 + 20 + 8 + 4) = 14, and the third is 12 + 8 + 18 + 20 + 8 + 4 = 70.

EXAMPLE 2 When sending packet data over ATM using AAL5, in which a whole number of cells each with 5 o header and 48 o payload is used, the last 8 o of the last cell of a packet being taken up with the AAL5 trailer, the first number is 65 535, the second is 48 – 8 = 40, and the third is 5 + 8 = 13. The segmentation and reassembly process is not considered to be fragmentation, partly because it is invisible to the destination application but also because it would be unhelpful to set the maximum size of a data unit as low as 40 o.

EXAMPLE 3 For a technology in which the packet header varies from 1 o for a data unit of size 0 o to 15 o to 3 o for a data unit of size 256 o to 4 095 o, and there is no inter-packet gap, the first number is 4 095, the second is 1, and the third is 1.

In the case of a route which traverses multiple links, the values shall be derived from the records for all the links, the first number being the smallest of the first numbers, the second being the largest of the second numbers, and the third being the largest of the third numbers.

EXAMPLE 4 For a route that traverses all three of the technologies in Examples 1 to 3, the first number would be 1 472 (the smallest of 1 472, 65 535, and 4 095), the second 40 (the largest of 14, 40, and 1), and the third 70 (the largest of 70, 13, and 1).

NOTE 1 The first number shows the largest data unit size supported by the route.

NOTE 2 The second and third numbers provide an indication of the smallest data unit size that can be transmitted efficiently. Together with the information in the Delay IE, they allow an application to choose an appropriate packet size for a live media flow, balancing the increase in efficiency when sending larger packets against the increase in latency waiting for enough data to fill the larger packet.

NOTE 3 The specification of the third number allows for cases, as in Examples 2 and 3, where the number of overhead o is different for different data unit sizes. The record does not attempt to model details of the encoding, so it does not show that in Example 2 the next smallest size above 40 o is 88 o, nor that in example 4 the second and third numbers do not refer to the same technology. It also does not report the overhead for a zero-length packet, because it might not be a good indication of the overhead incurred by small nonempty packets.

A PathMTU IE contained in a FindRoute request message shall apply to the route up to and including the link on which it is transmitted. A PathMTU IE contained in a FindRoute response message shall apply to the whole route. A PathMTU IE contained in any other message shall be treated as invalid.

NOTE 4 It is assumed that the parameters reported by this IE are the same in both directions on any link.

5.6.27 Number of destinations

The fixed part of a DestCount IE shall consist of a single unsigned integer. There shall be no variable part.

NOTE The length is not specified, and there is no requirement to use the minimum number of octets. In many cases one or two octets will be enough, but for some applications (live broadcast of a sports event, for instance) a larger number will be needed, and the sender may choose to always use four octets. There may be applications (for instance, distributing global information in ubiquitous sensor networks) in which even four octets is not enough.

5.6.28 Selector for an individual destination

A DestSelect IE selects a destination of a multicast and is in the context of a network element on the route of that multicast. The context may be the destination, or any network element through which the route passes on its way to the destination.

The fixed part shall be coded with a “selector” as specified below. There shall be no variable part.

An “arc” shall be coded as one octet containing *n*, followed by *n* o containing a value which identifies a network element adjacent to the network element which is its context. The coding of this identification is private to the network element which is its context, but shall remain valid as long as the link between the two network elements is operational, even if the route has been cleared down.

In the context of the destination, the selector shall be of zero length.

In the context of a network element through which the route passes on its way to the destination, the selector shall consist of an arc which identifies the next network element through which the route passes on its way to the destination, followed by the selector in the context of that network element.

5.6.29 User data

The fixed part of a UserData IE shall contain a data unit. There shall be no variable part.

NOTE Because there is no variable part, this format supports data units up to 65 535 o long.

5.6.30 Extended IE types

The fixed part of an ExtendRel IE shall begin with a RELATIVE OID rooted at 1.3.6.1.4.1. The fixed part of an ExtendAbs IE shall begin with an OID. In either case, the format of the remainder of the fixed part, and the semantics and protocols associated with the message, shall be specified by the owner of the OID.

NOTE 1 This is intended to allow proprietary and experimental IE types to be carried in standard messages and to be uniquely identified.

NOTE 2 ExtendRel IEs provide a more efficient coding in the case of OIDs descended from the "enterprises" node.

NOTE 3 There is no explicit specification of the length of the OID. The recipient of a message containing an Extend IE can match each OID it supports against the start of the fixed part of the IE. The only way this could be ambiguous is if the owner of an OID has allocated two message types, both descended from that OID, such that one is a subset of the other.

ExtendRel IEs for the same RELATIVE OID that are directly contained within the same message or IE shall be adjacent.

ExtendAbs IEs for the same OID that are directly contained within the same message or IE shall be adjacent.

6 Protocols

6.1 General

The unit transmitting a signalling message is referred to in this standard as the “sender”. The message shall be sent to a specific other unit which is referred to as the “recipient”. In the case of a message transmitted on a point-to-point link, the recipient shall be the unit at the opposite end of the link from the sender. Otherwise, the recipient shall be a specific unit connected to the same link and selected by an address, channel number, etc., as appropriate for the technology used on the link.

NOTE 1 A signalling message arriving at a network interface is always processed locally. In a switched network it is never forwarded to another interface by the routing fabric.

The recipient of a valid signalling message in which the most significant bit of the first octet is 0 (herein referred to as the “original” message) shall send either an “acknowledgement” or a “reply” or both to the sender of the original message. An acknowledgement should be sent if a reply will not be sent in time to prevent the original message being repeated.

An acknowledgement has the same message class and message type as the original message and the most significant bit of its first octet is 1. Its fixed and variable parts shall be as specified in 5.5.

NOTE 2 The specifications in 5.5 are intended to convey the minimum information necessary to identify the original message.

A “normal” reply has the same message type as the original message and the most significant bit of its first octet is a 0. Its message class is coded with a value 1 more than the message class of the original message. A reply shall not be sent unless a reply message is specified in the appropriate message type subclause of this Clause 6.

A ClearDown request message may also act as a reply to a message which refers to any route or flow that it clears down.

The sender of the original message shall treat the acknowledgement or reply as an indication that the message has been received and has been (in the case of a reply) or will be (in the case of an acknowledgement) processed. A unit shall not send an acknowledgement if it is possible that the message could be lost (for instance, through lack of buffer space) while awaiting processing in the protocol stack unless such a loss would be accompanied by a reset of the unit.

NOTE 3 The sender of an acknowledgement therefore takes responsibility for further processing of the transaction of which the message is a part.

The sender of a signalling message for which no acknowledgement or reply is received after a timeout shall repeat the message. After a longer timeout, or after a specified number of repetitions, the sender shall abandon the attempt to send the message and take appropriate action. The length of the timeout(s), number of repetitions, and action to be taken on abandoning the attempt to send the message depend on the technology used for the link and are therefore not specified in this standard.

NOTE 4 “Appropriate action” for a FindRoute will usually be to proceed as if a ClearDown had been received. “Appropriate action” for a ClearDown will usually be to complete the disconnection of the route or flow unilaterally. The failure to receive a reply might be treated as an indication that the link, or the intended recipient unit, has failed, but it is also possible that an error in the message, or in the other unit’s interpretation of it, could cause it to be ignored.

The recipient of an invalid signalling message shall silently ignore it.

The recipient of a signalling message where the context indicates that the message is a repetition (and thus has already been processed) should reply with an acknowledgement but otherwise ignore the message.

EXAMPLE 1 An incoming FindRoute request for a route of which the unit already has a record can be assumed to be a repetition if it comes from the same sender as the original FindRoute request. If it comes from a different sender it is not a repetition but indicates either that two different paths from the caller have converged at the unit or (if the hop count in the RouteMetric IE is significantly larger) that the request has gone around a loop.

EXAMPLE 2 An incoming ClearDown for a set of routes and/or flows that do not exist may be assumed to be a repetition. The routes, etc. having been cleared down when the original was received.

EXAMPLE 3 If an incoming FindRoute request is a repetition of one to which a FindRoute response was sent in reply, an acknowledgement is sent. Although the repetition suggests the response message may have been lost, the response is not repeated at this stage. It is left to the normal timeout mechanism to repeat the response later if no acknowledgment or reply is received for it.

6.2 Establishing a route

6.2.1 Connection of flows

The procedures specified in this subclause 6.2 may be used to establish a route on which there are no flows (it may still carry EndToEndData messages, and may have flows added later, as specified in 6.4), or flows may be connected at the same time as the route is established. In the latter case, synchronous flows shall be connected by messages passing in the same direction as the flow, while asynchronous flows shall be connected by messages passing in the opposite direction. A SyncAlloc or AsyncAlloc IE shall be present in the variable part of the FlowDescriptor IE that describes a flow if, and only if, the flow is connected by the message.

NOTE 1 The specifications for the two kinds of flow are different because synchronous flows require information (such as a layer 2 multicast address, and traffic parameters) from the source whereas asynchronous flows require information (such as a layer 2 unicast address) from the destination. In each case, if information is required from the opposite end this can be provided in the preceding message in the sequence.

NOTE 2 A SyncAlloc or AsyncAlloc IE of zero length can be used if the technology used on the link does not require any of the information the IE would carry.

A route may be established using FindRoute request and response messages, or may also require a confirmation message. Flows may be connected by any of these messages, or by a completion message.

NOTE 3 Synchronous flows away from the caller and asynchronous flows towards the caller can be connected by the request message, but it will often be more efficient for them to be connected by the confirmation message as this avoids setting up flows on branches of the path that do not reach the destination. Synchronous flows towards the caller and asynchronous flows away from the caller can be connected by the response message but in some circumstances (for instance to avoid sending paid-for media to the caller before the confirmation message has been received) will be connected by a completion message.

The process of connecting or disconnecting a flow is specific to the technology used for each link, and is additional to the actions specified in this standard.

6.2.2 Request message

To establish a route, an end unit, which is referred to here as the “caller”, shall send one or more FindRoute requests. If more than one is sent, they shall all be sent to different recipients. The route is then “pending” until further action is taken, as specified in 6.2.4.3.

The message shall include sufficient information to identify the called party.

NOTE 1 This will usually be a CalledAddress IE but can, for instance, also be a ServiceName or SourceName IE.

The message may include a CallingAddress IE.

NOTE 2 In the case of a point-to-point bidirectional call, a server can simply send replies on the call on which the request was received, without needing to know the client's address. The server might restrict the access it provides to anonymous callers, or reject all such calls. A destination can join a multicast without the source needing to know its address; again, the source can choose to prevent anonymous destinations joining the call (see 6.2.3.1).

The message shall include a FlowDescriptor IE for each flow that is to be connected on the route, except where the existence of the flow is implied by other parts of the message and the caller has nothing to say about it, for instance where the call is to “take” a source in whatever format the called party chooses to transmit. The FlowDescriptor IE may be directly contained in the message or contained in an Alternatives IE, as specified later in this subclause 6.2.2.

The flow reference may be zero, to indicate that it shall be chosen by the called party, unless the called party requires explicit flow reference values.

The IEs permitted or required inside a FlowDescriptor IE depend on the flags in its fixed part. For a synchronous flow, SyncParams may be included. For an asynchronous flow, AsyncParams may be included. For flows towards the caller (which is also the owner of the route identifier) SyncParams and AsyncParams specify the maximum values it can support; for flows in the opposite direction they specify parameters of the data stream it intends to transmit, and if AsyncParams is absent for an asynchronous flow a network-dependent default shall be assumed. Inclusion of SyncAlloc and AsyncAlloc is specified in 6.2.1.

NOTE 3 The following paragraphs specify other IE types that are permitted inside a FlowDescriptor IE.

One or more DataType IEs may be directly contained in the message and/or contained in any of the FlowDescriptor IEs. A DataType IE shall apply to the route if it is directly contained in the message, to the flow otherwise. Where there are multiple DataType IEs, each shall specify a different aspect of the data format, encapsulation, protocol, etc. Although this standard does not require any DataType IEs to be included, the called party may refuse the call if the message does not adequately specify information (such as the data format) that is conveyed by DataType IEs.

One StartTime IE may be directly contained in the message, where it applies to all flows, or some or all of the FlowDescriptor IEs may each contain one StartTime IE which applies to the individual flow. If present, then (provided the network supports delayed set-up) the flow(s) should not be set up in the switching fabric until shortly before the indicated time.

NOTE 4 This enables to reserve capacity ahead of time for scheduled events such as live outside broadcasts.

One EndTime IE may be directly contained in the message, where it applies to all flows, or some or all of the FlowDescriptor IEs may each contain one EndTime IE which applies to the individual flow. If present, it indicates that after the indicated time the flow may be cleared down if any of the resources it uses are required for other calls, and if all flows have been cleared down the route may be cleared down also.

NOTE 5 This can enable a flow to use resources which have been reserved for a flow that is scheduled to be connected later.

IE types 8 to 13 inclusive may be included both directly contained in the message, where they apply to the route, and in FlowDescriptor IEs, where they apply to the flow.

One or more Charge IEs may be directly contained in the message, showing the maximum the caller is willing to pay.

One RouteMetric IE with status 0 or 1, and/or one with status 2, may be directly contained in the message.

One PathMTU IE should be directly contained in the message.

One DestSelect IE should be directly contained in the message if, and only if, the call is to join a multicast for which the approval of the source is, or might be, required.

NOTE 6 The DestSelect IE, if included, will be empty (see 5.6.28).

Where the caller can accept several alternatives for the value conveyed by an IE of a particular type, for instance to receive at a lower quality if there is not enough bandwidth to support a higher quality signal, it may replace the IE with an Alternatives IE which contains several IEs of the relevant type, one for each alternative. In the case of a value conveyed by more than one IE (for instance, multiple DataType IEs specifying different aspects of the format plus a SyncParams IE specifying the resources required to carry that format), the IEs specifying each alternative may be collected together in a Group IE, and the Group IEs collected into an Alternatives IE. There may be more than one Alternatives IE, in which case they are separate sets of alternatives to be applied orthogonally (for instance, one for the called address and another for the data format and flow descriptors). When the message is passed on (see 6.2.3), any alternatives that cannot be supported (for instance, needing more capacity than is available on the link) shall be omitted.

6.2.3 Action on receiving a FindRoute request

6.2.3.1 Joining an existing multicast in a switch

A switch receiving a FindRoute request for a multicast which already passes through it shall take action depending on the “openness” of the multicast (see 5.6.20).

If the addition of a new destination requires the approval of the source, the switch shall pass the request (modified as required, e.g. with RouteMetric and DestSelect IEs updated) on to the next network element in the direction towards the source, except that if the request does not contain a DestSelect IE the switch shall not forward it but shall instead send a ClearDown in reply, as specified in 6.3.

Otherwise, the switch shall act as the responder as specified in 6.2.3.3.

If the source requires to be informed of the total number of destinations, the switch shall also send a NetworkData request containing a DestCount IE (see 6.5.2) to the source, no more than five seconds after the flow is connected. There should be an interval of at least two seconds between any two such messages for the same route.

NOTE It will usually not be appropriate to send the message to the source until the FindRoute confirmation message arrives. The two second interval limits the amount of load when multiple changes occur for the same multicast within a short space of time.

6.2.3.2 Other cases in a switch

A switch receiving a FindRoute request which is not for a multicast that already passes though it shall store any information that will be needed for processing subsequent messages, and pass on a copy (modified as required, e.g. with RouteMetric, Delay, and PathMTU IEs updated, and with an extra arc added to the DestSelect IE, if present) to one or more recipients, except that if there are no suitable recipients it shall send a ClearDown in reply, as specified in 6.3.

NOTE 1 An acknowledgement might need to be sent as specified in 6.1, except in the case where a ClearDown request is sent immediately.

NOTE 2 The information needing to be stored will include the sender of the incoming message, the recipient(s) of the outgoing message(s), and in each case the class of the most recent message (at this stage a request) and in the case of a response whether the offer was interim or final (see 5.6.25).

NOTE 3 Charge IEs may be added or removed, as appropriate, for instance an IE referring to a charge paid by a user to an ISP will be removed when the ISP passes the message on. The ISP may also add a Charge IE referring to peering charges, etc.

NOTE 4 The mechanism for deciding whether to pass the message on, and if so to which recipient(s), is not specified by this standard. It might use network-specific legacy protocols such as DNS, SIP, ARP, or PNNI. A new standard for exchanging network topology information might be developed using some of the message types which are shown as “reserved” in Table 3.

6.2.3.3 Action by end equipment

If an end unit receiving a FindRoute request can accept the call, it shall send a FindRoute response in reply. The unit is then referred to as the “responder”. Otherwise it shall send a ClearDown in reply, as specified in 6.3.

The FindRoute response message shall contain the same IEs as in the request (though they may be in a different order), with the following exceptions.

a) Any Alternatives IE, shall be replaced by one of the IEs directly contained in it, except that if the chosen IE is a Group IE, the Alternatives IE shall be replaced by the IEs directly contained in the Group IE. In general, any tentative data (e.g. flow references coded as zero) shall be replaced with actual data.

EXAMPLE If there is an Alternatives IE containing two or more Group IEs each containing a DataType IE and a SyncParams IE, the Alternatives IE is replaced by a DataType IE and a SyncParams IE, both coming from the same Group IE. If there are other DataType IEs directly contained in the request message, the chosen DataType IE will need to be adjacent to them in the response message, even if that is not where the Alternatives IE was.

b) Charge IEs from the request message shall be removed. Charge IEs shall be added to show the actual charge that will be levied when the confirmation is received. Charge IEs in the response do not need to correspond to Charge IEs in the request message.

c) Any RouteMetric or Delay IE with status 0 or 2 shall be removed. Any RouteMetric or Delay IE with status 1 shall have its status changed to 3. Any of the following IEs may be added (but not more than one of each): RouteMetric with status 0 or 1; Delay with status 0 or 1; RouteMetric with status 2; Delay with status 2.

d) Any FlowDescriptor IE that connected its flow shall be removed. Within a FlowDescriptor IE, a SyncParams or AsyncParams IE shall indicate the actual values to be used; these may differ from the values in the request message. A SyncAlloc (for synchronous) or AsyncAlloc (for asynchronous) IE shall be added if, and only if, the flow is to be connected by the response message.

NOTE 1 In the case of a synchronous flow that is connected by the response message, the requirement to add a SyncAlloc IE applies even if the flow is already being transmitted on the link traversed by the message.

e) Additional FlowDescriptor IEs may be included, for instance for a flow which is implied by a DataType or ServiceName or SourceName IE but for which no FlowDescriptor IE was included in the request message.

f) If the call was to "take" a multicast, an McastRoute IE shall be added, showing the path identifier and openness assigned to the multicast by its source. There is no correlation between the path selectors in the two identifiers, but the flow reference values shall correspond. If necessary, the flow reference values in FlowDescriptor IEs shall be changed to those used in the call indicated by the McastRoute IE. (For instance, it may be clear from embedded DataType IEs which flow is which, or there may only be one flow.) However, the “direction” flag in the low bit of the first octet of the FlowDescriptor IE shall still be coded as appropriate for the path in the fixed part of the message (e.g. set to 1 for a flow towards the caller).

g) Any DestSelect IE in the request shall be omitted, except where the responder is the source of a multicast that requires approval.

NOTE 2 Where the responder is the source of a multicast that requires approval, switches on the part of the route that is already carrying the multicast can use the DestSelect IE to route the response message and therefore do not need to remember the caller's route identifier.

h) Any IEs of type 3, 5-13, or 15 directly contained in the message that are unchanged from the request message should be omitted, likewise IEs of type 5-13 inside FlowDescriptor IEs.

NOTE 3 It is assumed that anything that requires the information from these IEs will have collected it from the request message.

6.2.4 Action on receiving on a FindRoute response

6.2.4.1 Messages including a DestSelect IE

When a unit other than the caller receives a valid FindRoute response message which includes a DestSelect IE, if the first arc identifies a network element to which the multicast identified in the McastRoute IE is already being transmitted, it shall remove the first arc from the DestSelect IE and pass the message on to that network element, modifying it as specified in 6.2.4.2.

NOTE Removing the first arc changes the context to the recipient.

Otherwise it shall remove the DestSelect IE and process the message as specified in 6.2.4.2.

6.2.4.2 Messages not including a DestSelect IE

When a unit other than the caller receives a valid FindRoute response message which does not include a DestSelect IE, if the unit has no record of the route, or has already sent a final offer (a FindRoute response message without an InterimOffer IE) for the route, or has received a FindRoute confirmation message for the route, it shall send a ClearDown request in reply.

Otherwise, it shall pass the message on towards the caller, modified as specified below.

If the message is an interim offer, or there are other recipients of the request message from which a final offer (or a ClearDown request) has not yet been received, and at least one interim offer for the route has already been sent towards the caller, the unit may store the message for a limited time (the limit not being specified in this standard) in the expectation that other offers will be received, and, at any time before the expiry of the time limit, choose one offer from all those that have been received, passing that offer on towards the caller and replying with a ClearDown request to the others.

NOTE 1 Passing on the first offer towards the caller without delay minimises the time to make the connection in the case where the caller will accept the first offer it receives. Delaying the other offers can reduce the number of messages required.

The message that is passed on shall be modified as follows.

a) If the request was passed to more than one recipient, and either the response is an interim offer or there is at least one other recipient from which a final offer has not yet been received, an InterimOffer IE shall be added containing the unit’s EUI-64 and a serial number from which the unit can identify the sender of the response message. Otherwise, no InterimOffer IE shall be added and the unit shall remember the sender as the sender of the final offer.

NOTE 2 This information might be required when the confirmation message arrives, see 6.2.5.

b) IEs such as RouteMetric and Delay (except where the status is 3) shall be updated to add the values for the link over which the message is to be sent.

c) Charge IEs referring to charges to be paid to the operator of the unit may be added. Charge IEs referring to charges to be paid by the operator of the unit should be removed.

d) Link-specific IEs such as SyncAlloc and AsyncAlloc may need to be modified to apply to the link over which the message is sent.

6.2.4.3 Action of the caller

When a unit receives a valid FindRoute response message for a route for which it is the caller it shall proceed as follows.

- If the route is no longer pending, or for any reason the offer is unacceptable, it shall reject the offer by replying with a ClearDown request.

NOTE 1 Reasons for rejecting an offer could include unacceptable parameters such as a data format which the unit cannot process, or an offer which is in some sense “better” (e.g. shorter route) having already been received.

- Otherwise, if the message is an interim offer, or there are other recipients of the request message from which a final offer (or a ClearDown request) has not yet been received, the unit may store it for later processing (when the final offer arrives, or after a timeout).
- Otherwise, if the message does not contain any Charge IEs, nor any flows that are not connected by the response message, nor a RouteMetric or Delay IE with status 1, and no previous interim offers have been received, then establishment of the route is complete, the route is no longer pending, and no further action is required other than sending an acknowledgement and any actions specified outside this standard.

NOTE 2 Such actions would include link-specific actions required to connect a flow.

- Otherwise, a FindRoute confirmation as specified below shall be sent in reply, possibly after awaiting the user’s agreement that the offer is acceptable.

NOTE 3 User confirmation is particularly relevant in the case where the message contains a Charge IE. The mechanism for requesting user confirmation is application-specific and outside the scope of this standard.

A stored interim offer may be processed at any time, either to reject it (for instance because a better offer has been received) or to send a confirmation. When a confirmation of an offer is sent, all other offers awaiting processing should be rejected.

When a confirmation of an offer has been sent, the route is no longer pending.

The FindRoute confirmation message shall contain the same IEs as in the response (though they may be in a different order), with the following exceptions.

a) All IEs other than FlowDescriptor, Charge, RouteMetric, Delay, McastRoute, and InterimOffer shall be removed.
b) Any RouteMetric or Delay IE with status 0, 2, or 3 shall be removed. Any RouteMetric or Delay IE with status 1 shall have its status changed to 3. Any of the following IEs may be added (but not more than one of each): RouteMetric with status 0 or 1; Delay with status 0 or 1; RouteMetric with status 2; Delay with status 2.
c) Any FlowDescriptor IE for a flow that was connected by the response message shall be removed.
d) Within a FlowDescriptor IE, IEs of types 5-13 that are unchanged from the response message should be omitted. A SyncAlloc (for synchronous) or AsyncAlloc (for asynchronous) IE shall be added if, and only if, the flow is in the correct direction to be connected by this message.

NOTE 4 Whereas the responder can choose whether to connect a flow by the response message or wait until it sends the completion message, the caller has no discretion in the case of the confirmation message, because it is the last opportunity to connect the flow.

e) Additional FlowDescriptor IEs may be included, for instance for a flow which it would be inappropriate to connect before the route has been chosen. SyncAlloc and Delay IEs (for synchronous) or an AsyncAlloc IE (for asynchronous) shall be included if, and only if, the flow is in the correct direction to be connected by this message.
f) If there is a Charge IE in the response message, there shall be an identical Charge IE in the confirmation message, showing that the caller accepts the indicated charge. Otherwise no charge shall be levied. If there is no Charge IE in the response message, there shall be no Charge IE in the confirmation message.

6.2.5 Passing on a FindRoute confirmation

When a unit other than the responder receives a valid FindRoute confirmation message, it shall proceed as follows.

- If the unit has no record of the route, it shall send a ClearDown request in reply.
- Otherwise, it shall pass the message on towards the responder, as specified below.

If the request was passed to exactly one recipient, the response shall be passed to that same recipient. The response should not contain an InterimOffer IE with the unit's EUI-64.

NOTE In this case it is expected that the unit will not have added an InterimOffer IE to the response message. It may check for an InterimOffer IE with its own EUI-64 in the response message, and take appropriate action if one is found, or it may simply pass the message on without checking it.

If the request was passed to more than one recipient, then if the confirmation message contains an InterimOffer IE with the unit's EUI-64, the IE shall be removed from the message and used to identify the recipient to which the message is passed. If this process fails it shall clear the route down as specified in 6.3. Otherwise the message shall be passed to the sender of the response message that was passed on as a final offer. In either case, the unit shall send a ClearDown request to each of the other recipients of the FindRoute request for the route that has not already been sent a ClearDown.

In either case, any Charge IEs that correspond to Charge IEs which the unit added to the response message shall be removed, and Charge IEs corresponding to any Charge IEs which the unit removed from the response message shall be added. Also, if the unit added a Charge IE to the response message but no Charge IE corresponding to it is found in the confirmation message, then no charge shall be levied for the call and the unit may abort the call by sending ClearDown request messages towards both the responder and the requester.

6.2.6 Action of the responder on receiving a FindRoute confirmation

When a unit receives a valid FindRoute confirmation message for a route for which it is the responder, if the message does not contain any flows that are not connected by the confirmation message, nor a RouteMetric or Delay IE with status 1, no further action is required other than sending an acknowledgement and any actions specified outside this standard. Otherwise, it shall send in reply a FindRoute completion message which shall contain the same IEs as in the confirmation (though they may be in a different order), with the following exceptions.

a) All IEs directly contained in the message except FlowDescriptor, RouteMetric, Delay, McastRoute, and InterimOffer shall be removed.
b) Any RouteMetric or Delay IE with status 0, 2 or 3 shall be removed. Any RouteMetric or Delay IE with status 1 shall have its status changed to 3.
c) Any FlowDescriptor IE for a flow that was connected by the confirmation message shall be removed.
d) Within a FlowDescriptor IE, IEs of type 5-13 that are unchanged from the confirmation message should be removed. A SyncAlloc (for synchronous) or AsyncAlloc (for asynchronous) IE shall be added.

NOTE We assume the flow is in the correct direction to be connected by this message, because any flows in the other direction will be connected by the confirmation message as specified in 6.2.4.3.

e) Additional FlowDescriptor IEs may be included, but only for flows that are connected by this message. A SyncAlloc (for synchronous) or AsyncAlloc (for asynchronous) IE shall therefore be included.

6.2.7 FindRoute completion message

When a unit other than the caller receives a valid FindRoute completion message, it shall proceed as follows.

- If the unit has no record of the route, it shall send a ClearDown request in reply.
- Otherwise, it shall pass the message on towards the caller, modified as follows.
  a) IEs such as RouteMetric and Delay shall be updated to add the values for the link over which the message is to be sent.
  b) Link-specific IEs such as SyncAlloc and AsyncAlloc may need to be modified to apply to the link over which the message is sent.

When the caller receives a valid FindRoute completion message, it shall take no action other than sending an acknowledgement and any actions specified outside this standard.

NOTE Such actions include link-specific actions required to connect a flow.

6.3 Disconnection of routes and flows

6.3.1 General

Whereas the protocol for setting up a route involves messages passing along the whole length of the path between the caller and the responder, disconnection proceeds independently on each link. This makes the process more robust, simplifies clearing down a route from the middle (for instance, when a link breaks), and allows switches to reroute calls around a broken link without tearing down and rebuilding the whole route. It also simplifies the situation where a branch of a multicast is being pruned back from a destination that has ceased to receive it at the same time as another destination in the same part of the network is joining it.

When a route is cleared down, all flows that follow the route are also cleared down. Other routes for the same call are not affected; to clear down a call, each route shall be cleared down separately.

Alternatively, an individual flow may be cleared down along a route, leaving other flows in place. Again, to remove a flow from a call that has several routes the flow shall be cleared down separately for each route.

Clearing down of a flow or route may be initiated by any of the units through which it passes.

NOTE A single message can request clearing down of multiple flows and/or routes. This is useful in the event of a broken link.

6.3.2 ClearDown request message

A unit shall request clearing down of one or more flows and/or routes across a link by sending a ClearDown request message.

The variable part of a ClearDown request message shall contain one or more Route IEs, and shall request clearing down of the selected route and/or flows.

NOTE A Route IE with a variable part requests removal of the selected flow(s), leaving the route in place even if all its flows have been removed.

In the case of a multicast for which a FindRoute response has been received by the sender of the ClearDown, the Route IE shall show the source's route identifier, not the caller's.

Between sending a FindRoute response and receiving the reply, a unit shall be prepared to receive a ClearDown with either identifier and shall interpret the “direction” bit in a FlowDescriptor IE in the context of the identifier in the Route IE in which it is contained.

The message may contain one or more Cause IEs. A Cause IE directly contained in the message applies to all flows and/or routes in the message. A Cause IE directly contained in a Route IE applies to all flows selected by it, also to the route if it selects the route (i.e. if it has no variable part). A Cause IE contained in a FlowDescriptor IE applies to that flow. If there is no Cause IE applying to a route or flow, the cause shall be assumed to be “normal call clearing”.

The message should not contain any other IEs.

6.3.3 Action on receiving a ClearDown request message

A unit receiving a ClearDown request message shall send an acknowledgement in reply, and shall take whatever action is necessary to remove flows and clear down routes as specified in the message, including issuing a ClearDown request for other links traversed by the route as appropriate.

NOTE 1 This action is specific to the technology used for the network and the link.

NOTE 2 If part of a unicast route has been cleared down, or if, when part of a multicast has been cleared down, the remaining part has no source, or no destinations, it will usually be appropriate to clear the route down completely. A similar consideration applies to individual flows. However, in some circumstances, or for some applications, a different action may be preferable, for instance when the cause is a broken link to attempt to reroute around the break instead of clearing the whole route down.

The recipient of the message shall ignore any IE which specifies a flow or route which it does not believe is present on the link, or which it is already in the process of removing.

NOTE 3 This covers the case where the message is a repetition. It also avoids problems in the event that clearing down of a route is requested from both ends at the same time.

6.4 Adding new flows to an existing route

The procedure for adding flows to an existing route shall be as specified in 6.2 except that the message type shall be AddFlow instead of FindRoute, the route identifier in the fixed part of the messages shall identify an existing route, and the parts of the procedure concerned with establishing the route shall be omitted.

IEs which are used only for establishing the route, such as CalledAddress and CallingAddress, should be omitted; if present, they shall match the existing endpoints, though they are not required to be the same as the addresses used in the original FindRoute.

6.5 Information messages related to a route

6.5.1 General

NetworkData and EndToEndData messages shall be used to convey along a route information that is not part of a flow.

A unit receiving an EndToEndData message shall send an acknowledgement for it (see 6.1) and also proceed as follows.

a) If the unit has no record of the route, it shall send a ClearDown request for the route in reply (in addition to the acknowledgement).
b) Otherwise, if the unit is an endpoint of the route it shall process the IEs it contains and may send a reply.
c) Otherwise, it shall pass the message on to the next unit along the route.

A unit receiving a NetworkData message shall proceed as for an EndToEndData message except that in case c) it shall take any action specified elsewhere in this subclause 6.5, and also make appropriate changes to any RouteMetric, Delay, PathMTU, or DestSelect IEs present, before passing the message on.

6.5.2 Notification of the number of destinations

A switch shall generate a NetworkData request message containing a single DestCount IE to inform the source of a change in the number of destinations as specified in 6.2.3.1.

Before passing on a NetworkData request message containing a DestCount IE a switch shall update its own record of the number of destinations and add to the number in the IE the number of destinations reached via neighbours other than the one from which the message was received.

The message shall not be passed on until two seconds have elapsed since the previous NetworkData request message (if any) was sent for the same route. If further such messages for the same route are received during that time, only one shall be passed on.

6.5.3 Changing service parameters

Procedures for “cross-layer” communication between network elements and end equipment, for purposes such as adjusting service parameters (and corresponding media compression parameters) as appropriate, for changing network conditions, are for further study.

6.5.4 Out-of-band data

An application may use an EndToEndData message to send along a route information that is not part of any flow. The message shall include a UserData IE holding the data unit to be conveyed, and may include one or more DataType IEs which indicate how it should be interpreted.

A message from the source of a multicast may include a DestSelect IE, in which case it shall be delivered only to the destination selected; otherwise it shall be delivered to all destinations.

An application shall not send an EndToEndData request message if less than two seconds have elapsed since a previous EndToEndData request message was sent for the same route.

NOTE EndToEndData is intended for occasional protocol messages between application entities in the endpoints. It is not intended to be used instead of connecting an asynchronous flow. See also 6.6.

6.6 Quasi-connectionless service

6.6.1 General

The quasi-connectionless asynchronous service conveys “packets” which are asynchronous data units with additional encapsulation as described in A.4.3.

This subclause specifies a message to support connection of a new uplink flow. Other aspects of the service, including the syntax and semantics of the encapsulation, are not specified in this standard.

6.6.2 Request message

An AsyncSetup request message shall contain a UserData IE containing the packet and one or more IEs to identify the destination.

NOTE The latter will often be a CalledAddress IE, but can, for instance, also be a ServiceName IE.

Other IEs, such as a DataType IE, should not be used because the information they contain will be lost if the message is forwarded on an existing uplink path. There may be provisions for carriage of such information in the encapsulation of a set-up packet, in which case it is inside the UserData IE.

6.6.3 Action on receiving request

If the recipient of an AsyncSetup request message is its destination, it shall connect the flow and process the packet as if it had been received on an existing uplink flow.

If a switch receiving an AsyncSetup request message has an existing uplink flow connected to the required destination, it shall forward the packet on that flow. Otherwise, it shall forward the message to one or more adjacent network elements, or, if there is no route towards the destination, send a ClearDown request in reply.

NOTE 1 As with a FindRoute request (see 6.2.3.2), this standard does not specify the mechanism for deciding to which adjacent network element(s) to forward it.

NOTE 2 It is expected that the recipient will also take other actions specified outside this standard, including link-specific actions to connect the uplink flow and a downlink flow to the caller, both towards the sender of the message and towards network elements to which the message is forwarded. If there is more than one of the latter, packets on the uplink flow would need to be multicast. If the switching technology does not support that, only one copy of the message can be forwarded.

NOTE 3 AsyncSetup messages are acknowledged as required by 6.1, but there is no other form of direct reply apart from the ClearDown.

7 Media formats

7.1 Identification

Encapsulation formats specified in this clause shall be identified by object identifiers rooted at the following location in the MIB tree:

```
mediaEncapsulation OBJECT IDENTIFIER::= { iso(1) standard(0)
             iec62379 network(5) signalling(2) encapsulation(3) }
```

Where definitions make reference to parameters, these parameters shall be interpreted as positive integer values which shall be appended to the object identifiers as additional arcs, in the order in which the parameters are listed.

NOTE These OIDs are appropriate for use in type 5 IEs to describe how the encoded media is encapsulated for transport across the network. OIDs for describing the coding of the media are specified in 4.1.2 of IEC 62379-2:2008 in the case of audio, and in IEC 62379-3 in the case of video.

7.2 Packet data

7.2.1 General packet data

The object identifiers specified in this subclause may be used to identify a flow that consists of a stream of packets without specifying anything other than the size and frequency of the packets.

```
fixedPackage OBJECT IDENTIFIER::=
                       { mediaEncapsulation fixed(1) }

variablePackage OBJECT IDENTIFIER::=
                       { mediaEncapsulation variable(2) }
```

A stream of fixed-size packets transmitted at regular intervals may be identified by an object identifier consisting of `fixedPackage` with two parameters, of which the first is the number of data octets in each packet and the second is the number of packets per second.

A stream of variable-size packets transmitted at regular intervals may be identified by an object identifier consisting of `variablePackage` with two parameters, of which the first is the maximum number of data octets in each packet and the second is the number of packets per second.

Packets that are not transmitted at regular intervals may be identified by an object identifier as above but omitting the second parameter.

7.2.2 Other packet formats

Formats defined in other standards should be identified by OIDs related to those standards, for example, formats defined in IEC 61883 should be identified by OIDs rooted at 1.0.61883.

7.3 Pulse-code modulated audio

7.3.1 Rationale

The format specified in this subclause 7.3, which is developed from the format specified in IEC 62365, is appropriate for carrying uncompressed audio over networks where the overheads per data unit are small and low latency can be guaranteed. In these circumstances it can support applications that need latencies lower than can be achieved with legacy packet networks.

The sequencing octet (see 7.3.2) provides, in only 8 bit, detection of missing or duplicated data units and a sample-accurate timestamp which allows different streams to be aligned.

The subframe format includes two optional four-bit fields. One, specified in 4.1.3 of IEC 62365:2009, allows IEC 60958 audio to be carried transparently, and the other, specified in 4.1.4 of IEC 62365:2009, provides data protection appropriate to PCM audio.

7.3.2 Sequencing octet

An octet, formatted as shown in Figure 8, should accompany the audio sample data for each sampling instant.

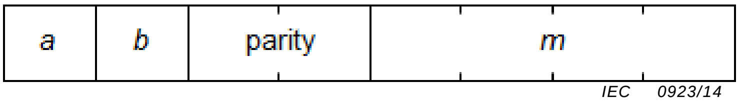


Figure 8 – Sequencing octet

In the following description

- *n* is the sample number modulo 3 072, in the range 0 to 3 071 inclusive, encoded as a 12-bit binary number;
- *k* is the least significant 6 bit of *n*;
- *a* is bit *k* of the "long" string specified below;
- *m* is the least significant 4 bit of *n*; and
- *b* is bit *m* of the "short" string specified below.

Starting at the most significant end of the sequencing octet, the first bit shall contain *a*, the second shall contain *b*, the third shall be such that there are an odd number of ones in the first three bits, the fourth shall be such that the total number of ones in the octet is odd, and the remaining four shall code the value of *m*.

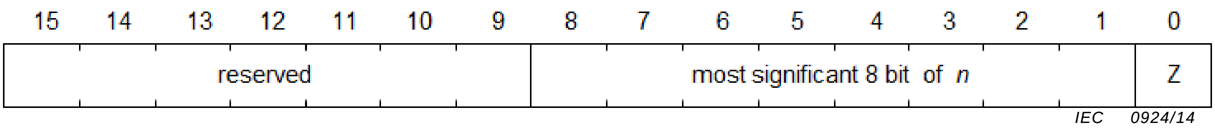


Figure 9 – "Short" bit string value

The "short" string shall consist of 16 bit numbered from 0 to 15 as shown in Figure 9. Bits 1 to 8 inclusive shall carry the most significant 8 bit of *n* (bit 8 being the most significant bit). Bit 0 shall be coded as 1 if bits 1 to 8 are all zero, 0 otherwise. Bits 9 to15 are reserved and should be coded as zero on sending and ignored on reception.

NOTE 1 It follows from the above definitions that $b = 1$ and $m = 0$ if, and only if, $n = 0$.

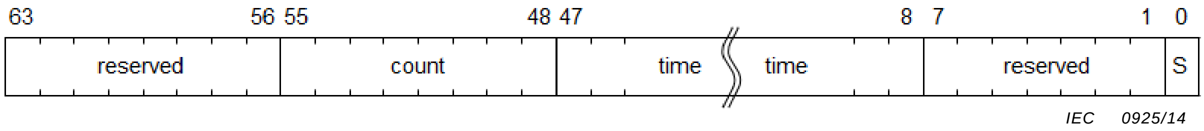


Figure 10 – "Long" bit string value

The "long" string shall consist of 64 bits numbered from 0 to 63 as shown in Figure 10, and shall be updated at the same time that *n* changes from 3 071 to 0. Additionally, bit 0 shall be set to 0 whenever *n* changes to a nonzero multiple of 64. The value, when transmission commences, shall be the same as if transmission had been continuing for the preceding two seconds.

NOTE 2 Using the description below, the value is not defined until the first update, and not fully defined until a further update at which the number of seconds is incremented. The above provision ensures that the sequencing octets seen by the recipient of a new multicast are well-defined and are the same as in the case where it has joined a multicast which other units were already receiving.

The value assigned when the string is updated shall be as follows. In the description, the "new" value is the value assigned and the "old" value is the value immediately before the update. The new value for bits 8 to 47 inclusive shall be the number of seconds that have elapsed since the sample-numbering epoch specified in AES53 (bit 47 being the most significant). If the new value in bits 8 to 47 is the same as the old, the new value in bit 0 shall be zero and the new value in bits 48 to 55 inclusive shall be one more (modulo 256) than the old value, interpreted as an integer with bit 55 the most significant bit. If the new value in bits 8 to 47 is not the same as the old, the new value in bit 0 shall be 1 and the new value in bits 48 to 55 inclusive shall be zero. Bits 1 to 7 and bits 56 to 63 shall be reserved and should be coded as zero on sending and ignored on reception.

NOTE 3 Bit 0 indicates "new second"; the first sample of a new second always has $a = 1$ and $n = 0$ and its "sequencing" octet is coded as 0xE0. This value in the sequencing octet does not occur at any other time.

NOTE 4 Assuming a sampling rate of more than 3 072 Hz, the new value in bits 8 to 47 when bit 0 is a 1 is 1 more than the old value. The value in bits 48 to 55 will not wrap round unless the sampling rate is more than 780 kHz.

NOTE 5 AES53 specifies that the number of seconds increments immediately before a sample whose number is a multiple of 3 072, so the value in bits 8 to 47 always corresponds to the number of seconds as specified by AES53. As noted in Clause A.3 of AES53, the number of samples in any one "second" will always be to a multiple of 3 072, but the long-term average will be equal to the sampling rate.

7.3.3 Subframe format

The subframe format shall be as specified in 4.1 of IEC 62365:2009, except that

a) if the sequencing octet specified in 7.3.2 is present and the protocol overhead specified in 4.1.4 of IEC 62365:2009 is included in the subframe, then the sequencing bit in the protocol overhead shall be replaced by a flag which is 1 if the subframe is valid, 0 if it should be ignored; and

b) the subframe is not required to be a whole number of octets.

7.3.4 Frame format

A frame shall consist of either

a) a sequencing octet followed by zero or more subframes, all of the same format, or

b) one or more subframes, all of the same format.

Frames with a sequencing octet and no subframes may be used to convey synchronisation information.

For each flow, the number of subframes, the format of the subframes, and whether the sequencing octet is included shall be the same in every frame.

There should be one subframe per audio channel, carrying samples taken at the same instant, with the *c*th subframe in a frame carrying the sample for channel number *c*.

NOTE 1 In some circumstances there will not be a one-to-one relationship between audio channels and subframes, see 7.3.6.5 and 7.3.6.6.

Option a) above should be used in preference to option b).

NOTE 2 Inclusion of sequencing information makes synchronisation of different signals easier and more reliable. The sequencing word specified in 4.1.4.1 of IEC 62365:2009 requires at least 8, and preferably 12, subframes per data unit, whereas on a network that supports smaller data units there may be only one or two. The sequencing octet specified in 7.3.2 carries more information than the IEC 62365 sequencing word, and is independent of the number of subframes per data unit, so provides a convenient way of conveying end-to-end sequencing information.

7.3.5 Transport

Each data unit shall contain one or more frames. Each frame shall be contained within a single data unit.

NOTE 1 The number of frames is chosen by the application. The maximum and minimum data unit size and overhead per data unit are reported in the PathMTU IE to aid its decision. The size of a frame is fixed for each flow, so the number of frames in a data unit can be deduced from the size of the data unit.

NOTE 2 Splitting a frame across several data units is specifically excluded. If the number of audio channels is such that a frame would be larger than the maximum data unit size, the channels can be divided into groups with each group being transmitted in a separate flow, and the groups being resynchronised by the receiving application using the information in the sequencing octet. If each flow has a different route identifier, the network can use multiple links where the total bandwidth required exceeds the capacity of an individual link.

EXAMPLE A bundle of 500 audio channels, sampled at 96 kHz with 32 bit per subframe, is to be sent over a 1 Gbit/s Ethernet network. The frame size is thus 2 001 o (including the sequencing octet), which is more than the standard Ethernet packet size, and the total bandwidth required (excluding overheads) is 1 536 Mbit/s, which is more than can be carried by a single link. Splitting the channels into two groups of 250 reduces the frame size to 1 001 o and allows the network to route the two groups over separate links. However, the bandwidth required for each group is 768 Mbit/s, so if less than 77 % of the capacity of a link can be allocated to synchronous flows it will not be possible to use groups of this size, and at least three will be needed.

7.3.6 Signalling of format

7.3.6.1 General

Encapsulation of pulse-code modulated audio as specified in 7.3 shall be identified by an object identifier consisting of `pcmAudioEncap` with the five parameters specified in 7.3.6.2 to 7.3.6.6 inclusive.

```
pcmAudioEncap OBJECT IDENTIFIER::= { mediaEncapsulation pcm(3) }
```

7.3.6.2 Synchronisation information

The first parameter shall be a value of the following type:

```
SynchronisationInfo::= INTEGER {
  absent          (0),
  sequencingOctet (1),
  iec62365        (2)
} (absent .. iec62365)
```

`absent` shall indicate that no synchronisation information is included in the data stream,

`sequencingOctet` shall indicate that the first octet of each frame is a sequencing octet as specified in 7.3.2.

`iec62365` shall indicate that synchronisation information is embedded in the subframes as specified in 4.1.4.1 of IEC 62365:2009.

7.3.6.3 Non-audio fields in subframe

The second parameter shall be a value of the following type:

```
additionalFields::= INTEGER {
  absent          (0),
  ancillary       (1),
  overhead        (2),
  both            (3)
} (absent .. both)
```

`absent` shall indicate that each subframe consists only of an audio sample word.

`ancillary` shall indicate that each subframe consists of an audio sample word followed by four bits containing ancillary data as specified in 4.1.3 of IEC 62365:2009.

`overhead` shall indicate that each subframe consists of an audio sample word followed by four bits containing protocol overhead, as specified in 4.1.4 of IEC 62365:2009 and as modified by 7.3.3.

`both` shall indicate that each subframe consists of an audio sample word followed by four bits containing ancillary data as specified in 4.1.3 of IEC 62365:2009 and four further bits containing protocol overhead as specified in 4.1.4 of IEC 62365:2009, as modified by 7.3.3.

7.3.6.4 Audio sample word length

The third parameter shall be the number of bits of audio sample data in each subframe.

NOTE This number might be greater than the bit depth signalled in the audio signal format, in which case IEC 62365 (like IEC 60958-4) requires the audio sample data to be padded with zeroes at the least significant end.

7.3.6.5 Subframes per frame

The fourth parameter shall be the number of subframes in each frame.

For PCM audio data, this should be the same as the number of audio channels signalled in the audio signal format. See also 7.3.6.6.

7.3.6.6 Frame rate

The fifth parameter shall be the nominal number of frames per second, rounded up to an integer.

For PCM audio data, this should be the same as the sampling frequency signalled in the audio signal format. Where a PCM audio channel is used to carry non-PCM data, as in IEC 61937, the frame rate may be unrelated to the audio sampling frequency.

Where the audio source and destination use double sampling frequency mode in IEC 60958 or AES10, a frame rate of half the sampling frequency, and a number of subframes which is twice the number of audio channels, may be used in order to avoid translating the channel status information, although this practice is to be discouraged as it prevents interworking with equipment using conventional encoding at the higher frequency.

8 Cause codes

8.1 ITU-T standard cause codes

Cause codes specified in ITU-T Recommendation Q.850 may be identified using OIDs rooted at the following location in the MIB tree:

```
causeCodeItu OBJECT IDENTIFIER::= { iso(1) standard(0) iec62379
             network(5) signalling(2) causeItu(4) }
```

The next arc shall contain a 7-bit cause value defined in Q.850 or a related Recommendation.

NOTE ITU-T Recommendation Q.2610 specifies additional cause values. Most are ATM-specific, but 73 (unsupported combination of traffic parameters) may be more generally applicable.

Where a cause value has diagnostics associated with it, each octet of the diagnostics should be appended as a separate arc.

EXAMPLE "Call rejected because the calling party address is not specified" is specified in 2.2.7.1.15 and 2.2.6.4 of ITU-T Recommendation Q.850 as cause no. 21 with 2 o of supplementary information, the first being 10000101_2 (decimal 133) coding “information element missing, permanent” and the second being 15, the IE identifier for “calling party address”, so the OID is 1.0.62379.5.2.4.21.133.15, coded in the Cause IE as relative OID 21.133.15.

Diagnostic information may also be provided in the variable part of the Cause IE, either in addition to, or instead of the additional arcs.

8.2 Other cause codes

The following location in the MIB tree is reserved as the root for other cause codes to be defined in future editions of this standard:

```
causeCode OBJECT IDENTIFIER::= { iso(1) standard(0) iec62379
             network(5) signalling(2) cause(5) }
```

Annex A
(informative)

Background

A.1 Support for future network (FN)

A.1.1 Rationale

Most current networks use IP (v4 or v6), but there is a growing recognition that this form of packet switching, which is appropriate for the irregular and unpredictable traffic generated by IT applications (except that IP provides communication between network interfaces, whereas applications usually require to access content, services, or equipment), has significant shortcomings for the “streamed” media which form an increasing proportion of the traffic carried on the Internet. For instance, ISO/IEC JTC1/SC6[2] are developing “Future Network” (FN) technology, which aims to meet the requirements of these and future services and overcome the deficiencies of the current IP based networks. FN, unlike “next generation” networks, is not constrained by a requirement to use the same routing techniques as the current Internet. This issue is also addressed in ITU-T Y.3001.

When FN is deployed it will need to co-exist and interwork with the previous technologies, for instance via gateways. To give the best performance, applications will need to be able to take advantage of improvements in Quality of Service (QoS) parameters such as latency and jitter, which are likely to be significantly lower on FN, and of new services which it may provide.

Some likely features of FN, and the way in which the protocols specified in this Standard support them, are described below.

A.1.2 Separation of route-finding from data-forwarding

FN is likely to support virtual circuit switching, in which routing decisions (which can be computationally intensive) are made by software before transmission of the data begins, and data forwarding (which, once the routing decisions have been made, is very simple) is handled by dedicated hardware. This reduces complexity in routers, and therefore also reduces the amount of power they consume. It also allows much more information about the transmission to be exchanged than could reasonably be accommodated in packet headers, and potentially allows the network to choose to switch some flows in the optical domain.

IP networks also separate discovery of the destination address, though not of the route to that address, from forwarding of the data, with protocols such as DNS and SIP. Also, MPLS can provide predefined routing within parts of an IP network.

The signalling protocols specified in this Standard provide for discovery of the route before transmission of the data begins, as in the call connection procedures that are used on circuit-switched networks, but the definition of a route (or rather, of a link: see Clause A.2 and Clause A.4) is broad enough that this can also apply to the discovery of the destination address on a IP network.

A.1.3 Mobility and resilience

The signalling protocols allow alternative routes to be discovered, and data flow to be switched between routes. This enables both failover when a link fails in a static network and handover when a mobile device moves from one cell or point of attachment to another.

2 ISO/IEC JTC 1/SC 6 Telecommunications and information exchange between systems

The level of resilience can be chosen to suit the application, for instance in the case of failover the new route may be found when the failure occurs, or a backup route may be chosen and kept in reserve until needed, or multiple copies of the data may be sent by different routes.

A.1.4 Addressing

No assumptions are made about the form of addressing used by the underlying network. The signalling messages do not have a fixed format (or fixed length field) for an address, so a very wide range of addresses can be used, including URLs. These addresses are independent of the addresses used for routing within the network, which will be dependent on the routing technology. Thus, processes such as DNS lookup take place entirely within the network, and are not visible to end-systems, which therefore do not need to be aware of any low-level addressing other than that used on the link or subnetwork to which they are connected.

As further detailed in Clause A.3, an address may include an *identifier* for a piece of equipment or content or a service, or a *locator* which shows where to find it, or both. There may be more than one locator, and the locators are then applied sequentially, in the same way as with a phone number containing a country code and area code. The call is routed first to the country and then to the area within that country.

A.1.5 Two classes of service

FN is likely to support two classes of service, “synchronous” and “asynchronous”. The synchronous service carries regular streams of data for traffic such as audio and video, whereas the asynchronous service is a “best effort” service for more bursty or irregular traffic, and can support both connection-oriented and connectionless routing.

The system works best if the underlying networking technology can reserve capacity for the synchronous traffic and use all the capacity not occupied by synchronous data (even that which is reserved but not used) for asynchronous data. This allows variable bit rate streams to be supported by simply making a reservation for the peak capacity required.

A.1.6 Negotiation and reporting of QoS parameters and data format

Provision is made for the application, or the source of the data stream, to specify the bandwidth required for synchronous data. This information can then be used to reserve capacity on the links traversed by the data, if reservation is supported by the underlying network.

The signalling messages report the end-to-end QoS to be expected. This allows an application to know, for example, whether to expect the incoming data packets to arrive at regular or irregular intervals, and allocate buffer space appropriately, a small amount to minimise latency in the first case and a large amount to minimise dropped packets in the second.

The signalling messages also describe other features of the data, such as encoding and encapsulation, so they contain all the information the recipient will need to decode the data. This can be used in a variety of ways, including negotiating the best compromise between quality of compressed media and capacity available in the network, also to support transcoding in gateways. The way in which formats and encapsulations are represented is fully extensible, providing a globally-unique coding not only for standard formats and protocols but also for those that are manufacturer- or application-specific or experimental.

Thus, some elements of the signalling messages (e.g. QoS parameters) are intended for the network, while others (e.g. encoding and encapsulation) are intended for the destination application. The network is expected to pass the latter through without interpreting them. For instance, it should not make assumptions about the service required based on the type of media being conveyed, as happens when “deep packet inspection” is used in IP networks.

A.2 Structure of the network

A network is composed of end equipment, switches, and links.

End equipment units convey media flows between media ports and network ports, the ports being part of the unit. Media ports may be physical interfaces, such as analogue or digital audio or video connectors, or internal sources or sinks of media flows, such as processing elements or memory.

A link connects network ports on different units together. It may be a dedicated point-to-point link, such as Ethernet over Cat5e cabling or a leased line connection over a telecommunications network, but it may also be a subnetwork (switched or shared-media) that connects more than two ports, or a larger network such as a switched telecommunications network or the current Internet.

Switches convey media (and other) data between network ports. A switch contains two “planes”: the data plane, which forwards incoming data to the appropriate output port (or ports, in the case of multicast), and the control plane, which processes the signalling messages specified in Clause 5.

The route followed by the media data from the source to the destination may need to traverse more than one kind of network. For instance, audio being transmitted from a studio in one location to a studio in another location will probably travel over the local infrastructure in both studios and a wide area network that links them. These three networks may well all use different technologies.

This standard specifies protocols which allow calls to be connected across heterogeneous networks which may use very different addressing schemes, and may be modelled in any level of detail, from individual switches and point-to-point links to a single “cloud” with no internal structure visible.

Different networking technologies also vary widely in the quality of service they are able to provide, from circuit-switched networks offering fixed latency and guaranteed delivery, through managed packet networks which can provide a certain level of assurance, to “best effort” packet networks (such as the Internet) which do not offer any guarantees at all.

A.3 Addressing

Addresses may perform one or both of two functions: location and identification. An identifier selects a particular object such as a physical unit, a service, or a piece of data. A locator shows where the required object is to be found.

Fixed-line telephone numbers are locators. They include a country code and an area code, which help the system to route the call to a specific telephone line. The call is answered by whoever happens to be near the phone at the time, and the caller then has to ask for the person they wish to speak to by name (which is, of course, an identifier).

The 48-bit MAC addresses used in Ethernet and other IEEE 802 networks (apart from group addresses and locally-administered addresses) are identifiers which uniquely identify a particular interface but do not contain any information as to where on the network the interface is to be found.

The 20-byte NSAP addresses used in ATM networks include a locator (the prefix) and an identifier (the ESI) in separate fields.

In IPv4 addresses (see IETF RFC 791) the subnet part is a locator and the host number is an identifier. It identifies an interface, though it is often used as an identifier for the piece of equipment that is connected through the interface. Unlike MAC addresses, the relationship between this identifier and a particular interface is often not permanent, and Network Address Translation means that an interface may appear to have different addresses in different parts of the network and several different interfaces may appear to have the same address.

The principal form of unique identifier for physical equipment used in this standard is the IEEE 64-bit extended unique identifier (EUI-64, see http://standards.ieee.org/develop/regauth/tut/eui64.pdf). An EUI-64 for any piece of equipment that has an interface which has a MAC address can be generated by inserting $FFFE_{16}$ into the middle of the MAC address. Identifiers that are not unique may also be used, for instance the name of a service may be used as the identifier for any piece of equipment that offers that service.

For calls within a subnetwork, only the identifier is required.

An address may consist of a series of addresses such that the call is routed to the first address, then from there to the next, etc. Each address is interpreted in the context of the equipment or location specified by the previous address, which thus acts as a locator. For instance, the address for an audio call may consist of the address of a piece of audio equipment followed by the identifier of a particular audio port on that equipment. The address of a piece of data may consist of its identifier (perhaps an IEC UUID or a SMPTE UMID) preceded by the address of a server (or group of servers) on which a copy is to be found. The address of a gateway (or even of a specific port on a gateway or switch) may be used as the locator for equipment accessed through that gateway. Such an address should, however, not be a unique identifier for the unit, because that would prevent calls being (re-)routed if the specified unit failed.

A.4 Calls, routes, paths, and flows

A.4.1 Definitions

A call is defined in IEC 62379-1 as conveying information either from a source unit to one or more destination units or (in both directions) between two units.

A path is a contiguous set of links along which the information might be conveyed from one unit to another. The term “path” is used when describing the process of discovering how to get from one unit to another across the network. A path may branch, in which case the different branches are different attempts at finding a way through. The eventual data flow will only be along one of the branches.

Once a set of links leading from one unit to the other has been discovered, it is called a “route”. A route may also branch, but only as required to deliver multicast data to several different destinations.

A route exists only in the control plane. It does not carry data, apart from signalling messages which are copied from one link to another (often being modified in the process) by software running in a computer which controls the unit through which they pass.

A flow is a single stream of information which is conveyed along a route.

A flow exists in the data plane. In the case of a circuit switched network, data units are copied from one link to another by the switching fabric with no intervention from the control plane software.

A call may carry several different flows. For instance, a connection between two studios may carry programme audio, talkback, and signalling (e.g. "on air" light) data as separate flows. Compressed data may be partitioned into a flow carrying a base layer and one or more flows carrying enhancement layers that improve the quality or add, for instance, surround sound channels. All destinations will receive the base layer, but the enhancement layer(s) would not be received by destinations that were not able to use the extra data, or did not have high enough bandwidth connections.

A call that needs high reliability (such as one that is part of the programme chain of a live broadcast, or in a safety-critical public address system) may have more than one route. For the highest reliability, all flows will be sent on two or more routes and the destination unit will use one copy and discard the other(s). Alternatively, a backup route might only carry the base layer, or it might be set up with no flows at all, with the flows only being connected in the event of failure of the main route.

A.4.2 Identification

Subclause 4.3 defines a structure for identifying calls, routes, and flows. This structure can be larger than would be the case if (like, for instance, ATM VCIs, or IP addresses and port numbers) it was transmitted with the data. Call identification consists of a globally-unique identification of the “owner”, which is normally either the originator of the call or the source of the data, and the “call reference” which is a 32-bit handle chosen by the owner. Thus, the call identifier is unique across the whole network. 32 bit are used so that the size of a call reference is unlikely to be a limitation on the number of calls that a unit can make, and the time before a call reference value is re-used can be made adequately long.

7 bit are used for the “route reference”, a handle on the route which a call follows. Most calls will only have one or two routes, but there will be cases where a call needs to set up new backup routes when existing routes fail and, again, the size of the field allows adequate time to elapse before handle values are re-used. Also, there is no particular benefit in shaving a small number of bits off this field.

The network is expected to ensure as far as possible that different routes for the same call do not pass through the same piece of equipment, which would then be a single point of failure.

The “flow reference” is a handle on the flow within the call, and is 24 bit. Again, most calls will only have a small number of flows, but this field has been made large enough that a call can carry a "tunnel" through which a large number of flows are routed (similar to an ATM VPC).

The flow reference is independent of the route, so flows that carry the same data by different routes can be easily identified. Where the same material is transmitted by different end units (for instance, where an audio input to the network is duplicated), the same call identifier (but, of course, different route references) may be used by the two units. If the two streams are identical (which can only be the case with digital inputs), they will use the same flow reference, but otherwise (for instance where the same analogue signal is digitised separately in each unit) they may need to use different flow references.

Note that a route has an existence independent of the flows that follow it. A route may be established without any flows, in which case signalling messages can still be passed along it. A flow exists in the routing tables of the data plane, whereas a route exists only in software records in the control plane. Thus, there are two distinct phases in connecting a flow: establishing the route, and setting up the switching fabric so that the data will be forwarded along it.

Also note that a flow passes through the same set of links and switches as the route it follows. This is different from the situation with SIP (IETF RFC 3261), in which the media flow will in many cases not pass through the SIP server(s) that connected it. See also 6.1.

A.4.3 Quasi-connectionless flows

HTTP clients often connect a TCP session, send a few packets, and then disconnect. They may do this multiple times for the same site. In an IP network, for the first connection DNS is used to discover the IP address, and a TCP SYN packet is sent. If other clients have recently accessed the same server, its address will already be in the caches of most of the routers through which the packet passes. They may also create cache entries for the sender's (i.e. the client's) IP address, which will mean there is no delay in routing the server's TCP SYN packet reply. For subsequent TCP sessions the addresses will already be in the caches throughout the route.

The protocols specified in this standard support a mechanism which provides equivalent performance in a connection-oriented network, using variants of the normal virtual circuits for asynchronous data which are referred to as “uplink” and “downlink” flows. They carry “packets” which consist of a data unit and some additional information (including flags to distinguish different packet types) as described below.

Uplink flows are many-to-one, and are intended for data from clients to a server. Data packets include a “handle” value (chosen by the server) to distinguish the different sources. They are administered by the control plane, which receives set-up request messages for flows that do not already exist and prunes connections that are unused. Route caching is achieved by not pruning them immediately. The control plane state is equivalent to a route, with each route carrying one flow.

Downlink flows are one-to-one, and are intended for data from a server to a client. They are connected and disconnected by additional data plane logic which interprets in-band signalling packets, and no information about them is held in the control plane, so they have no associated route.

The “handle” may identify the appropriate downlink route, in which case the server might not need to keep any state information (provided it is not using TCP); or it may identify a record which includes session state and authentication; or it could be a TCP port number. Note that an attacker can send a packet which appears to be from the client. In this case (unlike with IP) the reply will be addressed to the client, not the attacker. The server will still need to get an authenticated confirmation before implementing operations such as changing a password. Also, an attacker on a wireless network will be able to see the reply even if it is addressed to the client. The client's software should raise an alarm if it gets a reply to a request it has not sent, or sees another terminal sending a message purporting to be from it.

The data plane is assumed to be able to set up and clear down downlink connections without delaying the routing of the packet. The whole process is still expected to be less complex than routing of a connectionless packet.

In a typical scenario, a TCP session between a client and a server would proceed as follows.

a) The client sends a control plane message (similar to the message that would be used to set up a call) to the switch to which it is connected. The packet includes the server's address (maybe in the form of a URL) and a TCP SYN datagram, also any authentication information the server may require.
b) As the message propagates through the network (see Clause A.5), the switches set up an “uplink” flow (and route) from client to server and a “downlink” flow in the reverse direction.
c) When the message reaches a switch that already has an uplink flow to the server, the switch forwards it on that flow as a “set-up” packet. In subsequent switches the “downlink” flow is set up as the packet is forwarded.
d) When the server receives the set-up packet, it allocates a “handle” for the session, processes the TCP datagram, and sends on the new downlink flow an “initialisation” packet which includes the handle value and its reply to the TCP datagram.
e) Subsequent TCP datagrams are sent in “data” packets, those on the uplink flow include the “handle” value.
f) At the end of the session the server sends a “clear-down” packet which includes a TCP FIN datagram. Each switch that forwards this packet also clears down the downlink flow.

Uplink paths may also be used for UDP, or to tunnel connectionless network protocols such as IP across a connection-oriented network. In this case a simplified form of the set-up packet is used which does not set up a downstream flow. However, note that in the case of RTP setting up a call to carry a synchronous flow will give a better service.

A.5 Route finding

A.5.1 General

The process of establishing a route begins with the caller creating a new route identifier and sending a “FindRoute request” signalling message on one or more of its network ports. Each recipient checks whether it is the called party, in which case it responds to the request, or is directly connected to it, in which case it sends the request on to it. Otherwise it either rejects the request or sends it on via one or more of its network ports to units which are in some sense "closer" to the called party.

This standard does not specify how a switch chooses on which ports to forward a FindRoute request. The mechanism will usually be network-specific. Normally, a switch will forward the request on just one port rather than flooding it to all ports. However, if a loop is formed it can easily be detected, by comparing the route identifier in an incoming request with the route identifiers of requests that have already been forwarded, and this will occur before any data begins to flow.

As the request progresses through the network, it builds up a path which may have many tentacles reaching out towards the called party in different directions. When the request reaches the called party, a “FindRoute response” message is sent back along the path, and as it makes its way back towards the caller, the message accumulates "route metric" information which includes a measure of the "cost" of the path in terms of congestion and number of hops, and more literally the cost of the call if it passes over a public network. Each such response message is an offer to connect the route by the path over which it has travelled. The calling party chooses one offer and sends a “FindRoute confirmation” message which also causes all the other paths to be cleared down.

A ClearDown request may be issued at any time by any of the parties involved in the call. If issued by a switch it propagates in both directions. In the case of a multicast, in the direction towards the source, it stops when it reaches a branching point.

A ClearDown received in reply to a FindRoute request is a negative response which may be a "refusal" or a "backtrack". A refusal indicates that the called party has been found, but is not willing to accept the call, or that the called party's location has been found and the called party is not there. Any resources reserved for the call are relinquished as the refusal propagates back towards the caller. A backtrack indicates that the called party's location was not reachable via the path the request followed (maybe because the required resources are not available along that path, or because a link has been lost and the routing tables have not yet been updated to reflect the new topology), or that the called party is a service which was not available at the location the request reached, but may be available at a different location. A switch receiving a backtrack may send out a further request along a different path, or if all possible paths have been tried pass the negative response back towards the caller.

If the call is charged for, charging begins when the confirmation is received by the service provider. Inclusion of the new flow in routing tables, etc. may be delayed until the confirmation has been received, to prevent the user consuming the media without paying. Also, a call may be disconnected if no positive confirmation has been received within a specified time (which should be long enough for a user to reply to a "do you want to pay for this call?" alert).

The ClearDown request may be used as a negative FindRoute confirmation, for instance if the user does not wish to pay for the call or has got a better offer via another path.

A.5.2 Additional options

A switch may treat a positive response with a reduced capacity or high cost as a backtrack and try another path.

In any circumstances where a switch has forwarded a FindRoute request along more than one path, it propagates the first positive response, and any subsequent response that is a significant improvement on it, towards the caller. All such responses except the last include an "interim offer" indication which contains the switch's (EUI-64) identifier and a serial number. If the last response is not one that it would forward to the caller, it repeats the last response without the "interim offer" indication. A response message received by the caller may contain "interim offer" indications from several switches, one with no "interim offer" indications is final. A negative response is always final.

In the case where more than one response has been sent, the only possible course of action is to forward a positive confirmation along the correct onward path and send ClearDown requests on the other paths. The caller includes in the confirmation message all the "interim offer" indications from the response message, so that the correct path can be chosen at each switch.

A.5.3 Call to receive a multicast

The request message may include an upper limit on the capacity required and also a preferred and a minimum capacity. For instance, in the case of linear PCM audio, it might request 16-bit 96 kHz, but be prepared to accept 24-bit if that is what the source is outputting, or 48 kHz if the network is busy. Usually, the source will have advertised the formats it supports, so the caller can be reasonably sure what the options are. As the request message propagates through the network, link capacity may be reserved based on the parameters in the message (reserving the maximum at this stage, if available, else adjusting the message to reflect what is available).

The identifier for a multicast flow is owned by the source. The FindRoute messages use a temporary, "local", flow identifier owned by the caller. The response and confirmation messages include both identifiers.

Multicast calls include a specification (by the source) of the action to be taken by a switch that finds it is already carrying the flow requested by the caller. This action may be

a) to connect the caller to the flow without propagating the request on further,

b) to connect the caller and also inform the source that it has done so, or

c) to forward the request to the source.

Case c) will be appropriate for “private” flows where connection of the caller requires approval from the source. Except for case c), switches store some of the "route metric" information, including the latency, in case they later want to copy the flow to other callers. For parameters such as latency the value for the whole flow starting from the source, and maybe including processing delays upstream of the sending interface, is required, while for those that measure congestion, etc. the FindRoute response accumulates the value for the part of the flow downstream of the branching point.

A.6 Message format

The messages used to implement the protocol are structured in a similar way to those in ITU-T Recommendation Q.2931, with a fixed-format header followed by information elements in tag-length-value (TLV) form. As with a number of protocols and file formats that use TLV (including the ASN.1 coding used by SNMP), information elements may be nested within other information elements.

The TLV format makes it easy for the software in end equipment and switches to parse the messages, extracting the information they need and ignoring information which they do not need. The recipient of a text format, such as that used in SIP, scans the whole message to find the newline characters, removes white space, and recognises keywords, whether they are in upper case, lower case, or a combination of the two. This adds unnecessary complexity which may not be particularly onerous for PCs, etc., but can be more so for embedded code in interface units and also for switches that have to process a large number of messages.

The coding also provides various extension mechanisms that allow manufacturer-specific and application-specific information to be carried transparently, and to be identified unambiguously.

A.7 Media coding and encapsulation

To minimise the amount of transcoding required when transmitting audio and video over heterogeneous networks, and to increase the likelihood that equipment designed for different applications will interoperate successfully, this standard defines formats that can be used with any link technology.

The format for pulse-code modulated (PCM) audio is based on that specified in IEC 62365, which is in turn based on that specified in IEC 60958-4. Whereas IEC 62365 uses the fixed 48-octet cell size of ATM networks, the specification in 7.3 allows for variable package sizes. It supports networks in which it is efficient to send packages containing one sample per channel (to minimise latency) which, if the number of channels is small, as in mono or stereo, will be smaller than ATM cells. It also supports those with large per-package overheads (such as RTP/UDP/IP) for which larger packages should be sent.

Bibliography

IEC 60958-4, *Digital audio interface – Part 4: Professional applications*

IEC 61883 (all parts), *Consumer audio/video equipment*

IEC 61937 (all parts), *Digital audio – Interface for non-linear PCM encoded audio bitstreams applying IEC 60958*

ISO/IEC TR 29181-1, *Information technology – Future Network – Problem statement and requirements – Part 1: Overall aspects*

ISO/IEC 8824-1:2008, *Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation*

ITU-T Recommendation Q.2610, *Usage of cause and location in B-ISDN user part and DSS2*

ITU-T Recommendation Q.2931, *Digital Subscriber Signalling System No. 2 – User-Network Interface (UNI) layer 3 specification for basic call/connection control*

ITU-T Recommendation Y.3001, *Future networks: Objectives and design goals*

IEEE Standards Association *Guidelines for 64-bit Global Identifier (EUI-64™) Registration Authority* (http://standards.ieee.org/develop/regauth/tut/eui64.pdf)

IEEE 802 (all parts), *IEEE Standard for information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements*

IETF RFC 791, *Internet Prototol*

IETF RFC 3261, *SIP: Session Initiation Protocol* (IETF Proposed Standard)

AES10, *AES Recommended Practice for Digital Audio Engineering – Serial Multichannel Audio Digital Interface (MADI)* (Audio Engineering Society, New York, NY, USA)

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

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