Draft for discussion

WD 21559-x

Future Network — Protocols and mechanisms — Part x: Switching and routing

Source: UK expert

Contents

Page

Introduction

ISO/IEC TR 29181-1 describes the definition, general concept, problems and requirements for the Future Network (FN).

ISO/IEC TR 29181-3 examines the requirements for carrying data over digital networks, and identifies those that are not satisfied by the current Internet. It also notes some expected characteristics of new systems that are better able to satisfy the requirements, and specifies a model which supports both the existing system and the new systems. This will enable a migration to the new systems; it is also intended to make networks of all sizes easier to manage.

ISO/IEC 21558-x specifies an architecture which meets the requirements identified in TR 29181-3.

This International Standard specifies protocols and mechanisms for use within systems conforming to the architecture specified in ISO/IEC 21558-x.

The current text is based on a description of a prototype implementation; some additions are needed, marked by ed-notes in the text, including the following.

There are some parameters that are fixed in the prototype implementation but need to offer choices. Decisions are needed regarding some of them, and fields will need to be added to messages to signal their values.

1 Scope

This International Standard specifies protocols and mechanisms for use within systems conforming to the FN architecture specified in ISO/IEC 21558-x.

2 Normative references

To be added when the main body of text is completed; must include the "architecture" document (ISO/IEC 21558-x), ISO/IEC 62365, ISO/IEC 62379-5-2, and AES51. Or should we make an updated form of AES51 into a new sub-part of IEC 62379-5? Or simply specify the relevant information explicitly in this document (see EdNote at end of 5.3.1)?

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

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC TR 29181-1 and the following apply.

To be added when the main body of text is completed.

4 Abbreviations

To be added when the main body of text is completed.

5 FN physical links

Diagrams to be added (in all clauses)

5.1 General

An FN physical link conveys a sequence of frames. There shall be a whole number of frames per allocation period.

Where an FN physical link is implemented over a physical layer that was developed for another technology, the link partners may use the negotiation procedure specified below to manage the change from using that technology to sending FN frames.

FN physical links shall be full duplex.

Once the link is established, with FN frames being sent in both directions, it can be used to exchange signalling messages, which shall be used to convey any of the information necessary to set up IT flows that was not included in a negotiation process, and to align frames as specified in clause 7 below.

5.2 Frame format

5.2.1 Components of a frame

General provisions are specified here. Details are specified in an Annex which applies to a specific physical layer.

Each frame on a physical link shall include

timing field (see below)

slots for AV packets (see 5.2.2 and 5.2.3)

and may also include

trailing octets (see 5.2.3)

The timing field shall consist of four octets holding (big-endianly) a 32-bit value. The all-ones value shall show that no time is indicated. Time (modulo four seconds) shall be coded with seconds in the ms 2 bits and nanoseconds (in the range 0 to 999 999 999 inclusive) in the remaining 30 bits. Other code points are reserved.

NOTE 1: Coding time as counts of seconds and nanoseconds is compatible with PTP.

The timing field is a “framing” field. There may be other framing fields, e.g. to mark the start of a frame, to identify its position within the allocation period, or to detect transmission errors. A frame shall not contain anything that would require the recipient to buffer the data before processing it, except as provided in 5.2.3.

For each link, a “word” shall be defined as a whole number of octets. The number shall be strictly positive and shall be fixed for each format and physical layer.

A slot should consist of 64 octets. Each slot shall be a whole number of words.

The trailing octets (if present) shall be a whole number of words.

NOTE 2: On physical layers where the interface between MAC and PHY is typically 8 bits wide, such as 1Gb/s Ethernet, the word length can be a single octet. Where typical interfaces are wider, as with 10Gb/s Ethernet, the word length can match the width of the interface. The word length can be signalled in a framing field or during negotiation.

5.2.2 AV packets

Each slot shall either contain an AV packet or be an “empty slot”. In the case of a packet, the payload shall be any number of octets in the range 0 to 63 (inclusive). The header shall consist of one octet formatted as follows:

bit 7 (most significant bit): set such that the total number of bits set to 1 in the octet is odd

bit 6: flag *f*

bits 5-0: length, coded as the number of payload octets

The flag *f* shall not be used for routing, but shall be available for use by the higher layers; if it is used to guide reassembly of longer messages, it should be set to 0 in the last fragment of a message and 1 in others.

The frame format may provide for explicit indication of whether a slot is empty; otherwise, an empty slot shall be coded as a “null packet”, with length zero and $f = 1$.

5.2.3 IT data stream

The octets in each slot that are not part of an AV packet (after rounding the size of the packet up to a whole number of words), together with the trailing octets if present, shall be concatenated in the order in which they are transmitted to form the “IT data stream”.

NOTE 1: If the packet is not a whole number of words, the unused octets in the last word contain rubbish and are thus an overhead. Where the word length is more than one octet, this will always occur with zero-length packets (which use a whole word for one octet of information), so for frames with a longer wordlength a more efficient means of identifying empty slots can be useful.

The IT data stream shall carry IT packets and “idle” words. An “idle” word shall be a single word coded with a value that cannot be the first word of an IT packet header, and shall be transmitted whenever there is no IT packet available for transmission,

The recipient of an IT data stream shall interpret it according to the following three contexts:

(1) Searching: this shall be the initial context when the first frame is received, and shall be entered in the event of any error, including: frame of the wrong length; excessive gap between frames; parity error in the first octet of a slot; and error in integrity check fields for the frame or an IT packet header. There shall be a transition to "between packets" context when a sufficient number of consecutive "idle" words have been received.

NOTE 2: This International Standard does not specify how many is a "sufficient" number. It is a decision for the implementer.

(2) Between packets: in this context either an "idle" word or the start of a packet is expected. An "idle" word shall be ignored and the context remain as "between packets". A word that can be the start of an IT packet shall be interpreted as such and "within packet" context entered. Any other code point should be interpreted as an error, and "searching" context entered.

(3) Within packet: each word shall be interpreted as the next word of the packet. After the last word (as indicated by the payload length signalled in the header) the context shall revert to "between packets".

5.3 Negotiation packets

5.3.1 Payload format

Negotiation packets shall be formatted as specified in AES51, except as follows:

a) The Timing Information shall use the same coding as the timing field in frames (see 5.2.1).

b) For physical links, the additional code point hexadecimal 83, identifying a Link Confirm packet, shall be supported in the "type and format" octet.

c) The octet string specified as the Ethernet MAC client data may instead be sent as a service data unit for a service other than Ethernet, e.g. as a UDP datagram. The UDP port numbers to be used in this case are *tbd*.

NOTE: No semantics are assigned to the Timing Information in negotiation packets, so it will usually be coded as all-ones.

The packets described here are used for negotiation in the prototype implementation, which currently only supports 1 Gb/s Ethernet as a physical layer and always uses the Ethernet MAC layer when a link comes up. However, with the changes listed above and the IEs being almost all different (see below), very little of the original AES51 specification remains.

It would be better to use signalling messages, with one of the message types (and several of the IE types) that are currently reserved;. Then the same messages can be used for exchanging information on identity and capabilities on links which default to transmitting FN frames. An AES51 "type and format" code can be defined to encapsulate the messages on Ethernet links, or maybe an Ethertype (and also a UDP port number) can be allocated to them.

5.3.2 Information elements

5.3.2.1 Information element types

Information element types 01, 03, and 04 specified in AES51 shall not be used.

Information element type 02 specified in AES51 may be present for physical links able to carry AV flows, and if present shall be coded with the value zero. Use of this IE coded with a nonzero value is reserved.

As IE type 02 now only conveys one bit of information, it should probably be replaced by a bit in the "link type" IE, see 5.3.2.5.

Additional information element types specified in the following subclauses shall also be supported.

5.3.2.2 Sender's identifier

Information element type hexadecimal 82 shall be used to convey information relevant to higher-layer management, and shall be included if, and only if, the sender has a 64-bit identifier.

The first 8 octets of the information field shall contain the sender's 64-bit identifier, as specified in ISO/IEC 62379-5-2. Where the information field is longer than 8 octets, the remaining octets are reserved.

Currently the only coding specified in 62379-5-2 is an EUI64, in which the ls 2 bits of the first octet are always 00; some other formats have been defined and need to be included either in 62379-5-2 or here. See also Editor's Note to 5.3.2.4.

5.3.2.3 Flow label for signalling messages

Information element type hexadecimal 83 shall be used to specify the flow label to be used on signalling messages to the sender. The information field shall be coded with the value for the part of the packet header containing the flow label (including any protection fields), using the minimum number of octets and aligned at the high end if not a whole number of octets. If there is no type 83 IE, signalling messages shall use flow label zero.

NOTE: For all current link formats, the information field will be exactly two octets, containing the flow label and CRC.

5.3.2.4 Recipient's identifier

Information element type hexadecimal 84 may be used to issue a temporary 64-bit identifier to a unit which does not have a permanent one. It may also be used to confirm the link partner's identity. The first 8 octets of the information field shall contain the 64-bit identifier to be used by the recipient. Where the information field is longer than 8 octets, the remaining octets are reserved.

In the prototype implementation, a format with binary 10 in the ls 2 bits of the first octet is used; see Editor's Note to 5.3.2.2.

5.3.2.5 Link type

Information element type hexadecimal 85 shall be used to specify the type of link. In a Link Accept or Link Confirm the information field shall consist of four octets containing (big-endianly) a 32-bit record in the following form:

4 bits: protocol version: shall be coded with the value 1

4 bits: link type: 0 = physical link, 1 = virtual link, other code points reserved

22 bits: reserved: code as zero on transmission, ignore on reception

Need to allocate part of this reserved space to signal the size of an allocation period. In the prototype implementation (which codes the protocol version as 0 but is otherwise as here) it is always 16 frames (1ms); it could be coded as $\log_2$ (number of frames) with value 0 to 9 so in 4 bits.

Also should maybe add a bit that replaces IE type 02 (see 5.3.2.1 and EdNote in 5.3.1).

1 bit: 1 if timing information is to be exchanged, 0 if not

1 bit: 1 if FindRoute requests can be sent to the sender of the Link Request, 0 if not

In other packet types it shall consist of one or more 4-octet records in the same format, in decreasing order of preference, showing all the link types the sender can support. When requesting a physical link, all supported link types (physical and virtual) should be shown, with physical link types listed first.

NOTE: This allows a virtual link format to be used as a fall-back if the link partner does not support any of the offered physical link formats.

For each of the flags in the last two bits, if the bit is set in the Link Request it may be clear in the Link Accept, to show that the responder does not implement the facility.

If the penultimate bit is set for a physical link, the timing field shall be set as described in 8.1 in each frame; if not, it may be coded as all-ones. If the penultimate bit is set for a virtual link, both partners shall send link timing packets.

On a physical link or an internal virtual link, both bits should be set. On an external virtual link, they should be set to show what functions the client implements.

5.4 Link establishment procedure

5.4.1 Introduction

Link establishment occurs in the following three stages:

a) physical layer communication established

b) exchange of negotiation packets and/or exchange of identity etc information in signalling messages

c) exchange of synchronisation information and amalgamation of synchronisation domains

In stage (b), exchange of negotiation packets is not required if both sides default to sending FN frames, or one side defaults to sending FN frames and the other can detect that it does so.

On completion of stage (b), IT flows can be set up across the link. On completion of stage (c), AV flows can be set up across the link.

The negotiation procedure in stage (b) is outlined in the following subclauses, and details, including the encapsulation of negotiation packets, are given in the Annex that applies to the physical layer. Stage (c) is specified in clause 7.

5.4.2 Transmission of request

Negotiation of the use of the FN frame format on a network interface shall begin with transmission of a Link Request packet requesting a physical link as the first preference.

When a request has been sent, the interface shall enter “requesting” state.

Some of the states are defined in AES51; need to add references to them and definitions of the FN states.

We need to add some state diagrams, like figure 2 in AES51.

The Link Request message should be repeated if no reply is received. The number of repetitions and the interval between them are specified in the Annex that applies to the physical layer.

If there is no reply after the specified number of repetitions, and the interface supports virtual links over the network to which the interface is connected, it may attempt to establish one or more virtual links as specified in 6, otherwise it should enter the “passive” state.

5.4.3 Response to incoming Link Request

If a Link Request packet is received in "requesting" state, the sender's identifier shall be compared against the recipient's. If the sender's is larger, the packet shall be processed as if the recipient had been in "passive" state; otherwise, the packet shall be ignored.

NOTE 1: When a link between two FN network elements comes up, it is likely that the link partners will both send request packets simultaneously. Unlike AES51, we can't let the negotiation process happen in both directions, because of the requirement to co-ordinate switching to the FN frame format.

NOTE 2: If the link partners have identical identifiers, both packets will be ignored.

If there is no compatible format, a Link Reject packet shall be sent and "passive" state entered as in AES51.

NOTE 3: Having established that the link partner is an FN network element but there is no format that both support, there is no point in attempting to set up a virtual link, assuming the Link Request packet included all supported link formats.

Otherwise the actual configuration shall be transmitted in a Link Accept packet as in AES51, but the new state shall be "accepting" rather than "active".

5.4.4 Response to incoming Link Reject

As in AES51, on receiving a Link Reject packet in any state the recipient shall enter the "passive" state.

5.4.5 Response to incoming Link Accept and Link Confirm

The action for an incoming Link Accept packet shall be as in AES51 except that instead of entering "active" state the recipient shall send a Link Confirm packet and enter "ready" state. In "ready" state it shall be configured to send negotiation packets but receive FN frames.

When a Link Confirm packet is received in "accepting" state, the recipient shall switch to “transition” state.

In “transition” state it shall be configured to both send and receive FN frames, but transmission of the first complete frame might not have begun, so it cannot transmit any FN packets. As soon as FN frames are being transmitted correctly, it shall switch to “await sync” state.

NOTE: Implementers will usually find it convenient for outgoing frames on all interfaces to begin at the same time, so it can take up to one frame time before the first frame header is transmitted; an additional delay is then needed until the recipient's IT stream enters “between packets” state (see 5.2.3).

The link partner shall switch to “await sync” state either when it detects incoming FN frames or when it receives a signalling message; if the signalling message is a SyncInfo message with appropriate content, it shall switch directly to "aligning" state.

In "await sync" state it shall both send and receive FN frames, and can connect IT flows but not AV flows.

After switching to the “await sync” state, each unit shall send a SyncInfo signalling request message (see 8.2); it shall transition to “aligning” state when appropriate, as specified in 7.6. In “aligning” state, the phase relationship between incoming frames on the link and outgoing frames shall be measured, and the transition to “active” state shall occur when the phase has been established.

AV flows can only be set up in “active” state.

5.4.6 Errors and unexpected packets

Packets received in the wrong state may be either ignored or treated as errors. Note that, unlike AES51, a negotiation packet received when configured to receive FN frames will be seen as a format error.

Persistent errors in FN frames should cause "reset" state to be entered, which includes ceasing to send or receive FN frames.

6 Virtual links

6.1 Introduction

Whereas a point-to-point link between two FN-aware network elements transitions to a (single) physical link as soon as it is connected, a link to a legacy network can carry multiple virtual links, and these links can be set up at any time.

This International Standard does not specify how setting up of a virtual link is initiated (i.e. what causes a Link Request packet to be sent); typically it will be by a command from a control element or by the interface being configured to attempt it whenever the link comes up. Thereafter the process is similar to that for physical links, except that the legacy network's packet format continues to be used so the negotiation is as specified in AES51, i.e. without Link Confirm packets and without the action on receiving a Link Request depending on the sender's identifier.

Each virtual link has its own state, which shall be "requesting", "connected", or "terminating".

6.2 Transmission of Link Request

Setting up of a new virtual link shall begin with the sending of a Link Request packet, after which the link shall be in "requesting" state. If no reply is received, the packet may be repeated or the link may be terminated.

NOTE: This International Standard does not specify the frequency of retransmissions or how many there should be before giving up.

6.3 Response to incoming Link Request

A network element receiving a Link Request that it does not accept shall send a Link Reject if the request was unicast. If the request was broadcast, shall either ignore it or send a Link Reject.

A network element receiving a Link Request for a link which already exists with the same configuration and is in "connected" state shall reply with a Link Accept.

NOTE: This covers the case where a LinkAccept has been lost and also where both parties send a Link Request at the same time.

A network element receiving a Link Request for a link which already exists but with a different configuration and is in "connected" state shall either send a Link Reject or update the configuration to match the new request and send a Link Accept.

A network element receiving a Link Request for a new link which it accepts shall create the link with "connected" state and send a Link Accept.

6.4 Response to incoming Link Reject

A Link Reject packet on any link shall cause the link to be terminated. If the link was in "connected" state, a Link Reject shall be sent in reply.

6.5 Response to incoming Link Accept

tbd; I think there are additional states such as "await sync"

6.6 Errors, unexpected packets, and termination

If any packet other than Link Request is received for a non-existent virtual link, a Link Reject should be sent in reply.

If an encapsulated IT packet is received for a flow that is not connected, a ClearDown signalling message for the flow should be sent in reply.

Either party may initiate termination of a link at any time by sending a Link Reject and changing the link's state to “terminating”.

7 Network layer synchronisation

7.1 General

Frames on all links within an island shall be phase-aligned.

NOTE 1: This allows AV flows to be routed with minimal latency.

Frame rates for different islands within the same cloud shall be the same.

NOTE 2: This prevents queues building up in the de-jitter buffers.

NOTE 3: The phase-alignment of frames is completely independent of the synchronisation of digital audio and video; sampling rates of digital media do not need to be synchronised with the network, nor with other media flows .

7.2 Alignment of frames within an island

Each island shall be a single “sync domain”, within which all transmitted frames shall be phase-locked to the frames transmitted by a single unit, the “sync master”, so that the timing propagates out from that unit. The link on which a unit receives timing information is its “upstream” link. Any links over which it supplies timing information to a neighbouring unit are “downstream” links. All other links are “off-tree” links. The link partner on the upstream link is the upstream neighbour, and the link partner on any other link is a downstream neighbour.

NOTE 1: The links across which the timing propagates form a spanning tree, with the sync master at the root. These links are upstream in the direction towards the root, and downstream in the other direction. All other links are off-tree.

NOTE 2: The synchronisation method is further specified in the Annex that applies to the physical layer.

NOTE 3: End equipment may simply align the frame structure on the outgoing direction of its link directly with the incoming frames, for instance by starting an output frame each time the start of an incoming frame is detected. This is referred to as “loopback” timing. A link on which it is used is always on the spanning tree, with the end unit being a leaf.

When a link becomes a unit's upstream link, the point in the input frame at which an outgoing frame begins on each link shall be measured, and subsequent inter-frame gaps adjusted so that subsequent frames begin as close as possible to the same point. The average inter-frame gap shall be measured over each timing period as specified in the Annex that applies to the physical layer, and if the reference is lost the most recent average figure shall be maintained, using an algorithm such as a binary rate multiplier to minimise jitter.

NOTE 4: This reduces the impact on off-tree links until a new route to the sync master can be established.

7.3 Alignment of frames within a cloud

A cloud shall be a single sync domain, but frequency locked rather than phase locked.

NOTE 1: Each island will therefore have the same frame rate over time, although there may be considerable wander in the phase relationship between different islands.

Each island shall have its own sync master to which frames are phase locked; for one island this shall also be the sync master for the cloud, while for the others it shall have an upstream link which is a virtual link. Its upstream neighbour may be any unit in the neighbouring island; it is not required to be the island's sync master.

NOTE 2: There may be virtual links within an island, for instance if the island covers two sites which are connected by a physical link and also by a virtual link which is used as a back-up.

This case needs more thought. Within-island virtual links need a special status so that an AV flow will only be routed over one if it is necessary in order to avoid taking the same route as another path for the

same call. In that case the paths would have very different latencies, which would need to be taken into account when failing over from one to the other.

7.4 Establishment of sync domains

When a network element is powered on or otherwise reset, it shall form an island within which it is the sync master. When a physical link enters “await sync” state, the link partners shall exchange information regarding their sync domains. If they are in the same island, both will see the link as an off-tree link and they can set up synchronous flows as soon as they have discovered the phase relationship between the frames they transmit.

If they are part of different islands, the two islands shall be amalgamated as specified in 7.6.

NOTE: In this process, the one with the better reference (chosen in a similar way to the “best master clock” in PTP) is unchanged. The other is first reconfigured (if necessary) so that the unit connected to the new link is its sync master, and then the new link is set to being its upstream link.

A similar process shall be used when a virtual link is connected. If the two units are both in the same cloud, the link is off-tree. Otherwise, the two clouds shall be amalgamated with the new link being on-tree and the unit for which it is the upstream link becoming the sync master for its island.

7.5 Action on link down

If a link which is part of the spanning tree goes down, the unit downstream of it shall become sync master of a new sync domain.

NOTE 1: There is no change to any other links.

It shall freeze the frequency of the frames it generates at the value it had just before the link went down, and announce the new configuration to the units that are downstream of it in the tree. If any of them has an off-tree link to the previous domain, it shall act as if that link had just come up.

NOTE 2: This results in the tree being quickly reconfigured, before there can be significant drift in the phase relationship on off-tree links.

7.6 Procedures

Much of the following only applies to physical links; need to add text for virtual links.

7.6.1 Link up

7.6.1.1 General

If, when a link comes up, information in negotiation packets or elsewhere shows that the link partner uses loopback timing, the link can transition directly to the “aligning” state. Otherwise, it shall enter “await sync” state.

The unit shall send a downstream SyncInfo request message on the link immediately after switching to the “await sync” state. The normal process of repeating the message periodically until it is acknowledged shall apply.

7.6.1.2 Same domain

On receiving a downstream SyncInfo request message on the link containing the same sync source as its own (i.e. with the same “source of clock” value and, if it is less than 176, the same sync master's identifier), it shall switch to “aligning” state.

Need to check whether 176 is the right value.

7.6.1.3 Less “good” sync source

On receiving a downstream SyncInfo request message on the link in the case where its own sync source is “better” than the one in the message (i.e. has a larger “source of clock” value, or the same “source of clock” value and a larger identifier), it shall stay in “await sync” state until the link partner announces that it is ready for the link to transition to “on-tree”.

7.6.1.4 “Better” sync source

On receiving a downstream SyncInfo request message on the link containing a sync source that is “better” than its own (i.e. has a larger “source of clock” value, or the same “source of clock” value and a larger identifier), if it is not the sync master it shall first become sync master, as follows.

It shall forward the message on its upstream link, as an upstream message. Other units receiving the message shall forward it on their upstream links until it reaches the sync master. The sync master may ignore the message, for instance if it has already discovered a better sync source elsewhere.

If the sync master accepts the request, it shall send a handover message towards the new sync master. Each unit receiving a handover on its upstream link becomes sync master until it has passed the message on and the recipient has acknowledged it.

When the handover message reaches the unit that originated this part of the process, or immediately on receiving the SyncInfo request message if it is the sync master at that time, it shall switch to using the link as an upstream link, sending messages on all its links to inform the other units in its domain of the change, and to inform the link partner that the link is now a downstream link.

A unit receiving a handover message shall not announce the change to the rest of the network if it immediately passes the sync master rôle on to another unit; it shall, however, announce the change if it fails to pass it on.

7.6.2 Link down

When an off-tree link goes down, no action is required other than rerouting flows that traverse it. The same applies to a downstream link.

When a unit's upstream link goes down, the unit shall become sync master, setting the frame rate from the measurement in the most recent timing period as specified in 7.2, and send a downstream message on all its remaining links announcing that it is now sync master. If a unit receiving the message has an off-tree link, it should send an upstream message in the same way as in the “link up” case.

The link partner might also be in the detached part of the domain, so the unit should wait before sending the upstream message, to give the link partner time to pass on its own downstream message.

Some guidance as to the duration of the wait probably ought to be given; in the prototype implementation it is half a second, but a longer wait might be needed for larger networks. Note that we don't expect the framing to start to drift significantly for several seconds.

7.6.3 SyncInfo signalling messages

SyncInfo messages shall be signalling messages (as specified in ISO/IEC 62379-5-2) with message type 7 and the message class coded as “request”.

For information on frame timing the fixed part shall consists of two or thirteen octets, the first of which is the subtype and is coded as:

bit 7 (most significant): coded as 0 (code point 1 is used for network time, see 8.2)

bits 6-2: reserved, code as all-zero if not otherwise specified elsewhere

bits 1-0: message subtype: 1 = downstream, 2 = upstream, 3 = handover, code point 0 reserved

The second octet shall be a serial number which serves to associate an acknowledgement with the correct original message. The numbers shall not be interpreted as following any particular sequence.

The remaining octets shall not be present in handover messages.

The third octet shall contain the “source of clock” for the sync master for the cloud.

The coding needs to be specified; in the prototype implementation it is broadly similar to the codings in AES51. It applies to network time as well, where we say the ms bit is 1 if the external reference is still present, 0 if it's recently been lost

The next eight octets shall contain the sync master's identifier.

The last two octets shall contain an “uncertainty” value, which provides a measure of the uncertainty in the phase relationship between frames transmitted by the unit and the sync master's reference, i.e. the sum of the latency variation for all the links over which the spanning tree passes between the sync master and the unit sending the message. The first 3 bits shall contain the ones-complement of the number of virtual links, and the other 13 bits shall contain the ones-complement of the total number of nanoseconds of uncertainty for the physical links. If the sync master uses a global reference, the latter figure shall include the uncertainty in the relationship between the frames it transmits and the reference. A unit forwarding an upstream or downstream message received on a virtual link shall subtract 1 from the value in the 3-bit field unless it is already zero. A unit forwarding an upstream or downstream message received on a physical link shall subtract the uncertainty for the link on which it was received from the value in the 13-bit field, setting the field to zero if the result of the subtraction is negative.

NOTE: Coding these fields in ones-complement form means that larger values are “better” than smaller values. If the last eleven octets are considered to be an unsigned 88-bit integer, a “better” value is a larger value of that integer.

Where a virtual link is over a network that implements PTP it might be possible to synchronise the frames on the two islands quite closely, except that it won't be possible to react to a step change in the frame rate. It's tempting to say that if there's PTP with a global reference such as GPS available, then each island should lock its sync master to that, but I think we still need the spanning tree in place in case the reference is lost, unless we can be sure that all islands use the same PTP cloud and it'll remain coherent if it loses the GPS reference. Probably the best is to have the spanning tree defined, with a single sync master for the cloud sync'd to PTP, and sync the other islands up to PTP but revert to control via AES51 type 0x27 packets if the phase difference as measured using those packets gets too large.

Acknowledgements shall have a 2-octet fixed part which is the same as the first 2 octets of the message being acknowledged.

None of the messages shall have a variable part.

7.6.4 State information

For each link, a network element shall remember:

the header of the message that was most recently sent, in case it is not acknowledged;

whether the link is a downstream link (in which case it is in the same domain), and if not the information from the downstream message that was most recently received, which defines the link partner's sync master; and

whether AV flows can be routed across the link.

It shall also remember which (if any) is its upstream link; if there is an upstream link, its “current” sync master shall be the upstream neighbour's sync master, otherwise it shall itself be the sync master.

It shall also remember the information from the most recent upstream message that it forwarded, which defines a “new” sync master with whose domain its current domain might merge, and its upstream link for that domain. This information shall be deleted if that link goes down, or when it is converted into an upstream link.

7.6.5 Downstream messages

A downstream message shall report the sender's current sync domain. If received from the upstream link, it shall be forwarded to all the other links.

Any other link on which a downstream message is received can be assumed to be off-tree; if it shows that the link partner is in a “better” sync domain it shall be forwarded to the upstream link as an upstream message.

7.6.6 Upstream messages

An upstream message received on a downstream link shall be assumed to report a “better” sync master. If the receiving unit already has a domain (current or new) which is “better” than the one reported, it shall ignore the message (though it shall still send an acknowledgement); otherwise it shall forward the message on its upstream link.

If the recipient is the sync master, it shall reply with a handover message (in addition to the acknowledgement of the upstream message).

An upstream message received on the upstream link shall be acknowledged but otherwise ignored.

7.6.7 Handover messages

A handover message received on the upstream link shall hand the sync master role to the recipient. The sender shall continue to provide local timing until it receives the acknowledgement; after switching to taking its timing from the new upstream link, it should trim the phase relationship between incoming and outgoing frames on the link so that it is as close as possible to the value assumed when connecting AV flows.

NOTE: This reduces the likelihood that the phase will drift if connection and disconnection of links elsewhere on the network results in a link changing direction many times.

The recipient shall switch to local timing when it sends the acknowledgement. If it has no “new” domain, it shall stay as sync master and send a downstream message to all links. Otherwise, it shall forward the message towards the new sync master, switching the link to being an upstream link and taking its timing from it once it has received the acknowledgement.

A handover message received on a downstream link notifies the recipient that the link partner has joined its sync domain and the link is now a downstream link; this allows it to route synchronous flows over the link.

7.6.8 Transition to active state

When a link comes up and has reached “await sync” state, each partner shall send a downlink message informing the other of its domain. If both are in the same domain, the link is an off-tree link and both sides can proceed to “aligning” state. Otherwise, one of them will find that the other has a “better” domain and, once it has become sync master, send a handover message on the link. Unlike when sending a handover message on an upstream link, the sender shall switch to taking its frame timing from the link when it sends the message. The link partner can therefore switch to “aligning” state when it receives the handover message.

NOTE: If the unit in the domain that will change fails to become sync master, there will be some other change to its domain, which will be reported, including to the link partner, by a downstream message which supersedes the original downlink message, whereafter the process restarts using the domain reported in the new message.

The phase relationship established when a link enters the “active” state shall be maintained until it goes down, with the controls being trimmed if necessary on any occasion it is switched to being an upstream link.

8 Network time

8.1 General

Support for network time shall be optional.

Network time, if supported, shall be distributed across the network in a similar way to the phase-alignment of frames, though not (in general) by the same route. Network time may have discontinuities, for instance when changing to a different master.

Where network time is supported, the value transmitted in the timing field in frames on physical links (see 5.2.1) shall be the sender's network time at a defined point in the data stream. There will thus be a constant offset between the value in incoming frames and the recipient's time when the timing field is received, which is established by the exchange of protocol messages between the link partners.

NOTE 1: The offset is the sum of (a) the time between the sender sampling its timer and the timing field being transmitted, (b) the propagation delay, and (c) the time between reception of the timing field and the recipient sampling its own timer. The way in which each of these components of the delay is established is different for each physical layer technology.

In the prototype implementation, each of the three (and hence the total) is assumed to be the same in each direction. The specification for each physical layer needs to ensure that (a) and (c) are the same for all implementations, or alternatively that the signalling messages are enhanced so that the values can be reported; we also need to worry about whether (b) is the same in both directions. For Ethernet physical layers, a solution might be to exchange PTP packets (assuming PTP-enabled PHYs) before negotiating the change to sending FN frames.

Where network time is not supported, the timing field in frames on physical links shall be coded as all-ones.

NOTE 2: The most significant 4 bits of the number of nanoseconds can never be all-ones, so the all-ones coding can be recognised by checking those four bits.

NOTE 3: Network time is not mission-critical to the routing of AV flows, so there is no requirement for the entire network to use a single reference and no need for a "handover" protocol. Each unit advertises the source of its network time, and neighbouring units can choose to use it or not, depending on how it compares with rival sources and with their requirements (e.g. for media synchronisation).

NOTE 4: The system is intended to be able to distribute "real time" accurately enough to allow audio to be aligned to within the limits specified in AES11 (±5% of a sample time, which would be 130ns at 384kHz and about 1µs at 48kHz).

What are the equivalent limits for video?

8.2 Signalling messages

Information regarding network time is exchanged using SyncInfo messages. The fixed part is a total of 22 octets, of which the first (the subtype) is coded as:

bit 7 (most significant): coded as 1 (code point 0 is used for frame alignment, see 7.6.3)

bits 6-2: reserved, code as all-zero if not otherwise specified elsewhere

bits 1-0: message subtype: 1 = current source, 2 = new source, others reserved

and the second is a serial number; the sender can choose whether or not serial numbers for the two kinds of SyncInfo message (frame alignment and network time) come from the same number space.

The third to thirteenth octets describe the timing source in the same way as in frame alignment messages, except that the "uncertainty" value shall be coded as a single 16-bit field containing the ones complement of the maximum difference between network time as reported and the reference from which it was derived, in nanoseconds. An all-zero coding in this field shall indicate that the accuracy of the timing is either unknown or worse than ±65,534 µs. The most significant bit of the third ("source of clock") octet shall be set to zero if the

sender, or a unit upstream of it, is using a local reference which was synchronised with the reference indicated by the rest of the field but has since lost contact with the reference. The uncertainty value should be adjusted periodically to reflect the potential drift of the local reference.

The next 5 octets shall hold the “seconds” part of the time, as distributed by the indicated source, when the message was constructed. For audio synchronised to network time, this is the value that is coded in d47-8 of the “long string” specified in 7.3.2 of IEC 62379-5-2.

NOTE: The exact network time can be assembled from this value and the value in the timing field of frames on physical links.

The last 4 octets shall contain the sender's network time relative to the time in frames received from the link partner; if the two units are exactly synchronised as regards network time, the value will be the same as the offset that is added to the time in incoming frames to allow for latency on the link. The value shall be in nanoseconds, coded as a 32-bit unsigned integer, or all-ones if the information is not available.

8.3 Protocols

As with frame alignment messages, links can be classified as upstream, downstream, or off-tree, but these classifications have less effect on the protocol; in particular, the same messages are sent on an upstream link as on the other types. The links over which network time is distributed do not need to form a spanning tree; however, a unit shall not choose a source which might form a loop.

NOTE 1: A unit which loses its upstream link needs to wait long enough for the information to propagate around any loops before selecting an alternative source of timing that has the same reference and a larger uncertainty.

Before a unit makes a discontinuous change to its network time, it should notify all its link partners using a “new source” message reporting the proposed new timing. A unit receiving a “new source” message should take appropriate action (for instance, switching temporarily to an internal reference) before acknowledging it.

As soon as possible after a unit changes the source of its network time (other than a temporary switch to an internal reference that has been synchronised with the previous source), it shall notify all its link partners using a “current source” message. If a unit receives a “current source” message from its upstream neighbour containing different data from the previous one (ignoring the serial number, and taking account of any expected increase in the time reported) it shall notify all its downstream neighbours using a “current source” message.

NOTE 2: There is no maximum time specified between the change occurring and the “current source” message being sent, nor for how long a “temporary” reference can be used before it needs to be reported. Messages are likely to propagate down the tree significantly more slowly than the change in timing.

*Does this **only** apply to physical links, with PTP being used on virtual links?*

9 Management of network elements

9.1 Message format and protocol

Unit management messages shall be sent on a bidirectional call; the caller is a “management station” and the called party is the “managed unit”.

The messages shall begin with a 2-octet header formatted (big-endianly) as follows:

1 bit: 0 = request, 1 = response

3 bits: message type (0 = Get, 1 = GetNext, 2 = Status (see “Status reporting” below), 3 = Set, 4 = non-volatile Set, 5-6 reserved, 7 = console data (see below))

4 bits: for request: see below for GetNext, reserved (code as 0) for the others; for response: status (see below)

1 octet: sequence number (chosen by the sender)

For types 0-4, the remainder of the message shall consist of:

OID (coded according to ASN.1 BER, as in SNMPv1)

value (likewise; may be absent if not required, see below)

further {OID, value} pairs as above if reply to GetNext

Status values in responses shall be as follows:

0	normal reply
1	message has been truncated
2	no object corresponding to the requested OID
3	value is of the wrong type for the object, or otherwise illegal
4	object is read-only
5	other (unspecified) error, including message type not recognised
6	non-volatile Set not supported for this object
7	resource not available, e.g. can't create a new record because the table is full
8-13	reserved
14	temporarily unable to access the requested object
15	last Status Response message of cycle

NOTE 1: codes 0 to 5 are the same as in SNMP.

Except as described under “status reporting” below, the sequence number shall have no significance other than as a handle on the request to associate a reply with it. The recipient of a request shall simply copy the sequence number to the response message, without making any assumptions about sequence number values; in particular, it shall not ignore or reject a message which appears to it to be out of sequence.

The OID and the value shall form an SNMPv1 VarBind except that the enclosing SEQUENCE shall be omitted. The value should be omitted in the Get and GetNext requests; it shall be omitted in any response which cannot include a correct value, in which case no further OIDs shall follow.

NOTE 2: The OID tag is not omitted, so the third octet will always have the value 6.

Open Sound Control (in which arcs have names rather than numbers) supports “wild card” characters in the names; something similar ought to be supported; note that 0x80 isn't a useful value for the first octet of an arc number (it codes 7 leading zero bits) so could be used to introduce some kind of wild card notation; does ASN.1 say anything about use of 0x80 in the first octet of an arc?

Set (including non-volatile Set) asks to change the value of the object indicated by the OID to the value in the message. If the object cannot have the requested value, it should be set to the nearest possible value; for instance, if it can take values 0, 10, 20, ..., 100 and the request is to set it to 37 then it should be set to 40.

Non-volatile Set shall be processed in the same way as Set, but also written to non-volatile memory; commands in the non-volatile memory shall be processed on start-up. The value written to the memory shall be the value in the request, not the one in the reply, and shall replace any command for the same OID that is

already in the memory; if a non-zero status code is returned, nothing shall be written to the memory. Status code 6 shall indicate that the command has been processed as a normal (volatile) Set. A non-volatile Set with an OID but no value shall delete any command for that OID in the flash; the status returned shall be 0 if a command was deleted, 2 if none was found, otherwise 5 or 6.

A response message shall have the same message type and sequence number as in the corresponding request. For Get and Set the OID shall also be the same as in the corresponding request, and the value shall be the current value of the object with that OID if possible, otherwise the value shall be omitted (so the message ends at the end of the OID, or possibly earlier). Status code 1 (the SNMP “tooBig” code) shall indicate that the response has been truncated at the maximum message size; the VarBind should be truncated rather than being omitted altogether.

The response to a Set shall show the value of the object immediately after the change is made, so in the example above the value will be 40 (as specified in ISO/IEC 62379, though SNMP requires it to be 37, which is not helpful). If the value in the request is of the wrong type for the object (status code 3) or the object is read-only (status code 4), the response shall show its current value in the same way as a Get response.

If the OID in a request message is corrupt, or the message type is not recognised, a response should be returned which has status code 5 and is a copy of the message up to and including the point where the error occurs (but always at least 2 octets), for instance 2 octets long if the message type is not recognised, 3 if the 3rd octet is not the OID tag.

In the case of a GetNext, the second half of the first octet of the request shall be coded with 1 less than the maximum number of objects to be included in the response, or 15 to indicate “as many as will fit in the maximum MTU size”. The first object in the response shall be the “lexicographical successor” (as in SNMP) of the OID in the request, and each of the others shall be the “lexicographical successor” of its predecessor.

NOTE 3: This is similar to SNMP GetNext when the second half of the first octet is zero, GetBulk otherwise.

If there are no further objects in the MIB, the OID shall be coded as a single octet with value 127 (so that it is greater than any legitimate OID), and the value shall be omitted.

NOTE 4: This is different to the SNMP response which shows the requested (or preceding) OID and the special value “endOfMibView”.

If there is more than one OID in the reply, the status shall be 0; if an error occurs when accessing an object other than the first, the reply shall end with the preceding object, so that a subsequent GetNext can return the error code in a reply that does not include any other objects. The “end of MIB” OID shall not be regarded as an error, even if it is the only OID in the reply

NOTE 5: Thus, unlike SNMP, GetNext will not return the noSuchName status value.

9.2 Status reporting

The management station may request to be kept informed of changes in the state of the managed unit by sending a type 2 request which is only two octets long. The response shall also be two octets long, with the same sequence number as in the request.

Maybe we should allow individual objects to be monitored by including their OIDs in the request message, or by writing a table indexed by the OIDs. However, it may be preferable to multicast the status messages, which would work better if there aren't too many options. Or maybe there's a need for both unicast, tailored status reports and multicast, standard ones.

The management unit shall report its state in messages which are coded as a Status Response and include one or more (OID, value) pairs, up to the maximum MTU size.

The *n*th message of a cycle shall have sequence number *n*; if *n* reaches 255, it shall then wrap to 2, missing out 0 and 1. These messages are referred to as “in-cycle” messages.

NOTE: They never have sequence number 0 and only the first of a cycle has sequence number 1. The last message in a cycle can be just 2 bytes long; this will occur if all the objects it would have reported have ceased to exist since the previous in-cycle message was sent.

Objects which have changed shall be reported in “change” messages which have sequence number 0 and do not form part of the cycle.

9.3 Console data

Messages of type 7 shall carry data from the user at the management station in one direction, and text for display on the management station's screen (or incorporation in a log file) in the other direction. In each case the remainder of the message (from the third octet onwards) shall consist of UTF-8 text.

The response shall act as an acknowledgement; status values shall be as follows:

0 text received and passed on to the relevant process

1 text received but could not be passed on

2-15 reserved

The managed unit should send console output on the management connection on which a type 7 message was most recently received; when there is no such connection, data should be saved up in a buffer (discarding the oldest when the buffer is full) and output when the next type 7 is received.

9.4 MIB for call management

Objects used for call management shall be as specified in ISO/IEC 62379-5-1.

Annex A
(normative)

Links implemented over 1Gb/s Ethernet physical layer

A.1 FN physical links

A.1.1 Frame format

Each frame as transmitted over the MII shall be formatted as follows:

2 octets:	Ethernet preamble (hexadecimal 5555)
1 octet:	Ethernet SFD (start-of-frame delimiter, hexadecimal D5)
1 octet:	AES51 type-and-format byte (see A.1.2)
4 octets:	timing field
7744 octets:	121 slots for AV packets
40 octets:	trailing octets
4 octets:	Ethernet FCS (frame check sequence)

NOTE 1: See 5.2 for specifications that are not dependent on the physical layer.

The size of a word shall be one octet.

NOTE 2: With a single-octet word size, there is no wasted space and thus no significant benefit in providing an explicit indication of an empty slot, so the “null packet” coding is used for empty slots.

NOTE 3: The preamble is shorter than in standard Ethernet, but is believed to be sufficient in all circumstances for point-to-point links. Frames are longer than standard Ethernet frames, but within the limits for “jumbo” frames.

If there is no reference to which to align the frame structure, the inter-frame gap (during which “valid” is false) shall be 14 octets. Otherwise it shall be in the range 12 to 16 octets inclusive, and should be in the range 13 to 15 octets inclusive.

NOTE 4: The inter-frame gap as received can be different from the gap as transmitted, because gap bytes can be inserted and deleted where the byte stream crosses clock domains in the PHYs.

The FCS on transmitted frames shall be as for an Ethernet frame. The FCS on received frames should be checked and if incorrect logged as a transmission error, but processing of incoming frames shall not be delayed until the FCS has been received.

NOTE 5: AV and IT packet headers have their own protection. Waiting until the FCS has been checked before processing the frame would delay AV flows unnecessarily, and potentially result in discarding a whole frame when there is a single bit error.

I included the FCS because I thought there ought to be some kind of equivalent to SDH's various check bits, and the logic to calculate the FCS has to be present for when the interface is used for Ethernet (including the AES51 negotiation packets). It is not clear to me what proportion of errors would be detectable by other means (e.g. premature negation of “valid”, or an error reported by the PHY); if it's most of them, it might be better to replace the FCS with four more trailing bytes, especially if an FCS error is likely to cause entry to “searching” state long after the error has ceased to have any effect.

Also, really simple devices should be able to go straight to the FN format (see 5.4.1), in which case if the integrity check field is needed there ought to be an option to use something simpler to calculate, such as longitudinal parity.

A.1.2 Allocation and timing periods

A timing period shall consist of 16384 frames, numbered sequentially from 0 to 16383.

In the type-and-format byte, the most significant 3 bits shall be 011, the next bit shall be 1 if the frame number is a multiple of 512, 0 otherwise, and the least significant 4 bits shall contain the least significant 4 bits of the frame number.

NOTE 1: Thus the hexadecimal value in the type-and-format byte will always be in the range 60 to 70 inclusive.

NOTE 2: Timing periods are only used locally within a network element, so the format does not signal the frame's position with in the timing period.

The number, *n*, of frames in an allocation period shall be a power of 2, and the frame number of the first frame in each period shall be a multiple of *n*.

NOTE 3: Thus the longest allocation period supported by this format is 512 frames, nominally 31,98976 ms, and a frame with type-and-format value hexadecimal 70 will always be the first in a period. If an allocation period of 16 frames is used, the position of each frame in the period is shown by the type-and-format value. For longer allocation periods, the position can be established by counting frames, using the frames with code hexadecimal 70 as a reference.

A.1.3 IT data stream

An IT packet header shall be formatted as:

13 bits: payload length (octets) - 1

3 bits: CRC (see below) covering the payload length

13 bits: flow label

3 bits: CRC (see below) covering the flow label

Each CRC field shall contain the 1's complement of the remainder of the division (modulo 2) by the generator polynomial $x^3 + x + 1$ of the product x^3 multiplied by the preceding 13 bits.

NOTE 1: This is the same calculation as the protection for sequence numbers and audio samples in ISO/IEC 62365.

The payload shall be any number of octets in the range 1 to 2000 (inclusive).

NOTE 2: The value in the first octet of a valid header is therefore in the range 0 to 62 inclusive.

An "idle" word shall be coded with the hexadecimal value FF.

A.1.4 Negotiation packet addressing

This subclause will need revision if the changes discussed in the EdNotes to 5.3.1 are made; in particular, it currently assumes AES51 encapsulation, which would need to be specified here if removed from 5.3.1.

A Link Request packet requesting a physical link should be sent to destination address 01-80-C2-00-00-01.

NOTE 1: This is intended to ensure the packet is only seen by the link partner, so the link will be a direct point-to-point link that does not pass through any other equipment. The specified address is used for "pause" frames and does not pass through IEEE802 switches. Pause frames have Ethertype 8808 so the request packet should not cause any confusion to a switch. A link through a hub will be half-duplex, so will be recognised as unable to support an FN physical link.

All other negotiation packets shall be unicast to the address which was the source address of the incoming Link Request or Link Accept packet.

A.2 FN virtual links

A.2.1 Introduction

Virtual links as specified in this clause may also be connected over Ethernet links that are half-duplex, or operate at data rates other than 1 Gb/s.

A virtual link using Ethernet encapsulation shall be identified by the source and destination MAC addresses. A virtual link using UDP encapsulation shall be identified by the source and destination IP addresses and UDP port numbers.

All packets except the Link Request shall be unicast. The sender's address and port number from the Link Request and Link Accept packets shall be used as the respective addresses and port numbers.

A.2.2 Transmission of Link Request

The Link Request packet may be sent to the broadcast address, or (with Ethernet encapsulation) to the multicast address specified in AES51, or to a unicast address.

NOTE 1: The broadcast address is appropriate for a client running a management application over IP. A unicast address allows a link to be set up with a specific partner.

The Link Request packet shall have the same set of VLAN tags that its sender proposes shall be used for subsequent packets. The Link Accept packet may have a different set, and it is the set in the Link Accept packet that shall be used in subsequent packets.

NOTE 2: Each set may, of course, be empty. There is no restriction on the tags that may be used in a Link Reject.

Where UDP encapsulation is used, the destination port number shall be *tbd*.

The prototype implementation uses 35037, which is the same 16-bit value as the AES Ethertype, and there was an intention to register it as a well-known port number for AES protocols, but that hasn't happened so we'll probably need to get it registered for this standard.

For an internal link, the sender's port number shall also be *tbd*. On an external link the client should use a dynamic port number.

NOTE 3: Dynamic port numbers occupy the top quarter of the number space, though some operating systems seem to treat all port numbers above about 1000 as dynamic. Between two network interfaces there can only be one internal link but there can be multiple external links, for instance if a client is running several different programs, each of which has a connection to the FN network.

A.2.3 Encapsulation of IT packets

IT packets transmitted on a link shall use the same encapsulation as the negotiation packets that set the link up, except that the AES51 type-and-format octet shall be coded with the hexadecimal value 26.

Each Ethernet packet or UDP datagram shall carry a single IT packet. The IT packet header shall be coded as specified in A.1.3.

The recipient of an encapsulated IT packet shall ignore it if the IT packet header is invalid.

A.2.4 Link timing packets

A.2.4.1 Procedure

Link timing packets convey the phase relationship between FN frames transmitted by the link partners.

On an internal link, each link partner shall transmit link timing packets with a nominal interval of two seconds between transmissions.

Link timing packets may also be transmitted on external links. If link timing packets are not transmitted, keep-alive packets shall be transmitted at the same rate. A keep-alive packet shall have the same format as a negotiation packet, except that the type-and-format octet shall be coded with hexadecimal 84 and there shall be no IEs.

A.2.4.2 Format

Link timing packets transmitted on a link shall use the same encapsulation as the negotiation packets that set the link up, except that the AES51 type-and-format octet shall be coded with the hexadecimal value 27.

The timing field shall code the time of transmission in the same way as on a physical link.

Following the timing field shall be at least twelve octets, of which the first four shall carry (big-endianly) a 32-bit value recording the phase at the time of transmission and structured as:

3 bits: reserved, code as zero

16 bits: frame number

13 bits: octet number within the frame

The frame number shall count transmitted frames modulo 65536. The octet number shall count from zero.

NOTE 1: There is no specification of the point at which a new frame is reckoned to begin, because these packets are only used to measure drift in the phase relationship, and it is thus only the relative values that are important.

NOTE 2: There is no specification of the contents of the other eight octets, which are intended to be overwritten with the local time and frame phase when the packet is received.

A.3 Link-specific Information Elements in signalling messages

A.3.1 General

The following subclauses specify IEs which ISO/IEC 62379-5-2 specifies to be link-specific.

A.3.2 AsyncAlloc

The fixed part of an AsyncAlloc Information Element shall consist of two octets containing the value (flow label and CRC) to be coded in the third and fourth octets of the header of IT packets transmitted on the flow in the direction towards the sender of the message containing the AsyncAlloc IE. There shall be no variable part.

A.3.3 SyncAlloc

A.3.3.1 Physical links

For the purposes of the SyncAlloc IE, a frame shall consist of 122 slots numbered from 0 to 121. Slot 121 shall correspond to the octets between the last slot of the frame and the first slot of the following frame, and shall not be allocated to a flow.

NOTE 1: Each of slots 0 to 120 will be 64 octets. Slot 121 will be 64 to 68 octets, depending on the number of gap octets.

A flow's allocation of slots shall be specified as the location in the allocation period of the first slot and a table listing the other slots (if any). The first slot may be anywhere in the allocation period. The other slots shall be listed in order from the first slot to the end of the allocation period and then from the start of the allocation.

NOTE 2: When routing an AV flow, the outgoing allocation period can have any phase relationship to the incoming allocation period, so packets in the slot that comes earliest in the incoming allocation period might not be routed to the slot that comes earliest in the outgoing allocation period. However, packets from the “first” slot in the incoming SyncAlloc IE can always be routed to the “first” slot in the outgoing SyncAlloc IE, similarly the rest of the slots in the allocation.

I'm not sure whether to take this last sentence out of the Note and change “can” to “should” or even “shall”; it might be useful to be able to distinguish different slots in an allocation, as suggested in Note 3. On the

other hand, we don't want endsystems relying on packets turning up in particular slots, because a flow passing over a virtual link won't preserve the relationship.

The fixed part of a SyncAlloc IE shall consist of four octets describing the table format and the “first” slot, followed by the table. There shall be no variable part.

The first four octets shall contain (big-endianly) the following fields:

9 bits: frame number in the allocation for the first slot

7 bits: slot number within the frame for the first slot

4 bits: (length (in bits) of each element in the table) - 1

12 bits: number of entries in the table (which is 1 less than the number of slots)

Where the allocation period is less than 512 frames, unused most significant bits of the frame number may be used to indicate the position of the first slot within the longer period.

NOTE 3: Where an AV flow is routed from a link with a longer allocation period to a link with a shorter one, additional slots might be required on the second link. If the flow then passes to a third link with the same allocation period as the first, the higher bits can be used (in a way not specified by this International Standard) to establish which slots need to be routed and which will always be empty.

The table shall be coded as a number of elements of the indicated length packed big-endianly into the remainder of the IE, beginning at the fifth octet.

Each entry in the table shall consist of one or more elements, and represent the position of a slot as n, the number of unallocated slots (including slot 121 if relevant) between it and the previous slot, as follows: defining e as the largest value that an element can take, the entry shall occupy k+1 elements, with e in each of the first k elements and j in the last, where $n = (k * e) + j$ and $0 \leq j < e$.

EXAMPLE 1: The first three numbers in the first four octets are 0, 115, and 2, and the first 18 bits of the table are 371070 when represented in octal. The first slot is slot 115 of frame 0, and an element is 3 bits so $e = 7$. The first entry is 3, so 3 slots (116, 117, and 118) are skipped and the second slot of the allocation is slot 119 of frame 0. The second entry is 7 + 1 = 8, so 8 slots (120 and 121 of frame 0 and 0 to 5 of frame 1) are skipped, and the third slot is slot 6 of frame 1. Similarly the next three entries are 1, 0, and 7, and the next three slots are slots 8, 9, and 17 of frame 1.

EXAMPLE 2: An allocation period is 16 frames, the numbers in the first four octets are 5, 22, 7, and 7, and each of the remaining 7 octets contains the value 243. The first slot is slot 22 of frame 5, and the others are slot 22 of frames 7, 9, 11, 13, 15, 1, and 3. Thus the allocation is of 8 slots which are evenly-spread within the period.

NOTE 4: The number of elements in the table depends on the entry values, and is not signalled explicitly. The element length can be chosen such that the length of the table is minimised; this will usually be when most entries have a value between $e/2$ and e, as in Example 2 above. If the element length is 1, the table is a bit map, with a 0 for slots that are allocated and a 1 for slots that are skipped.

A.3.3.2 Virtual links

The fixed part of a SyncAlloc Information Element shall consist of a 32-bit signed integer holding the number of slots for AV packets within the allocation period.

An AsyncAlloc IE shall be sent in the opposite direction to specify the flow label to be used in the header of the IT packet within which the AV packets are encapsulated.

This assumes a one-to-one relationship between AV flows and the IT flows in which they're encapsulated. Do we need to support anything more complex? See Note 1 to the “Virtual links” subclause of the “AV services” clause (currently 9.5) in the Architecture draft.