ANNEX B CHANNEL ACCESS SUBLAYER (mandatory)

The channel access sublayer handles links between peers.

B.1. Channel Access Sublayer Service Definition

The Channel Access Sublayer provides services to the Subnetwork Interface Sublayer. These services are:

1. Execute requests by the Subnetwork Interface sublayer to "*Make"* and *"Break"* Physical Links

2. Notify the Subnetwork Interface sublayer of changes in the state of a Physical Link.

3. Accept S_PDUs (encapsulated in the appropriate primitive) from the Subnetwork Interface sublayer for transmission on a Physical Link.

4. Deliver S_PDUs (encapsulated in the appropriate primitive) received on a Physical Link to the Subnetwork Interface sublayer.

The term "physical link" is used in AComP 5066 to mean a logical link between a pair of nodes with consistent ARQ state.

In order to provide these services, the Channel Access Sublayer implements a protocol that specifies the tasks that must be executed and the rules that must be obeyed by the sublayer. While a number of different channel-access protocols are possible, the one that is suitable for this document is referred to as the Channel Access Type 1 (CAS-1) protocol, and described herein.

The Channel Access Sublayer also manages the operation of ALE links with peer nodes.

B.1.1. **Changes in This Edition**

The functional differences between this specification and Edition 3 are set out in Section B.10.

The following key changes have been made:

1. The CAS-1 protocol has been extended to enable negotiation of support of this version of the protocol. For peers not supporting this version of the protocol, only Edition 3 compatible features are used.
2. ALE support has been added. In order to achieve this, various models of peer access are defined, which has added some new early sections to this document and various protocol changes.
3. Hard Links. These are now deprecated and specified in Section B.9.1, rather

than in the core specification.

B.2. Interface Primitives Exchanged with the Subnetwork Interface Sublayer

The implementation of the interface between the Channel Access Sublayer and the Subnetwork Interface sublayer is not mandated or specified by this standard. Since the interface is internal to the subnetwork architecture and may be implemented in a number of ways it is considered beyond the scope of STANAG 5066. A model of the interface has been assumed, however, for the purposes of discussion and specification of other sublayer functions.

Despite the advisory nature of the conceptual model of the internal interface between the Subnetwork Interface sublayer and the Channel Access Sublayer, there are some mandatory requirements that are placed on any interface implementation.

The interface must support the service-definition for the Channel Access Sublayer, i.e.:

1. the interface **shall** enable the Subnetwork Interface sublayer to submit requests to change the state of a physical link, i.e., to make or break a physical link with a specified node address;

2. The interface **shall** enable the Channel Access sublayer to notify the Subnetwork Interface sublayer of changes in the status of the physical link;

3. The interface **shall** allow the Channel Access sublayer to accept S_PDUs from the Subnetwork Interface sublayer;

4. The interface **shall** allow the Channel Access sublayer to deliver S_PDUs to the Subnetwork Interface sublayer.

Additionally, the protocol-control information from the Subnetwork Interface sublayer that is required for the management of the Channel Access sublayer **shall** not be derived from knowledge of the contents or format of any client data or U_PDUs encapsulated within the S_PDUs exchanged over the interface.

[Note: user's that encrypt their traffic prior to submittal may use the subnetwork. Subnetwork operation must be possible with client data in arbitrary formats that are unknown to the subnetwork, therefore any service requirements or indications must be provided by interface control information provided explicitly with the user data.]

The Channel Access Sublayer does not queue traffic. The DTS will queue traffic and in some modes of operation SIS will queue traffic. CAS can process information with its peers and between the layers immediately, without need for any queueing.

The interface may use knowledge of the contents of S_PDUs (excluding the contents of any encapsulated U_PDUs) to derive protocol control information for the Channel Access sublayer. This approach is highly discouraged, however. The recommended approach for implementation is that information required for protocol control within the Channel Access sublayer should be provided explicitly in appropriate interface primitives.

In keeping with accepted practice in the definition of layered protocols, and as a means for specifying the operations of the sublayers that are mandated by this standard, the communication between the Channel Access Sublayer and the Subnetwork Interface sublayer is described herein with respect to a set of Primitives. The interface Primitives are a set of messages for communication and control of the interface and service requests made between the two layers.

By analogy to the design of the client-subnetwork interface, the technical specification of the Channel Access Sublayer assumes communication with the Subnetwork Interface sublayer using primitives prefixed with a “C_”. A minimal set of C_Primitives has been assumed that meet the requirements stated above and the general function for each C_Primitive is given in Table B-1. These C_Primitives are given without benefit of a list of arguments or detailed description of their use. As noted initially, they are offered for information only as a means of describing the interaction between the Channel Access sublayer and the Subnetwork Interface sublayer, and as general guidance for a possible implementation.

Table B-1- Nominal Definition of C_Primitives for the Interface between the Channel Access sublayer and the Subnetwork Interface sublayer (non-mandatory, for information-only)

NAME OF PRIMITIVE	DIRECTION (see Note)	COMMENTS
C_PHYSICAL_LINK_MAKE	SIS →CAS	request to make a physical link to a specified node
C_PHYSICAL_LINK_MADE	CAS→SIS	report that a physical link was made
C_PHYSICAL_LINK_REJECTED	CAS→SIS	report that a physical link could not be made and that the initiating request is
C_PHYSICAL_LINK_BREAK	SIS→CAS	request to break a physical link
C_PHYSICAL_LINK_BROKEN	CAS→SIS	report that a physical has been broken (by request, or unilaterally)
C_UNIDATA_REQUEST	SIS→CAS	delivers an S_PDU to the Channel Access Sublayer, requesting normal data-delivery
C_UNIDATA_REQUEST_CONFIRM	CAS→SIS	confirms S_PDU delivery to the remote node using the normal data-delivery
C_UNIDATA_REQUEST_REJECTE D	CAS→SIS	notifies that an S_PDU could not be delivered to a remote node using the
C_UNIDATA_INDICATION	CAS→SIS	delivers an S_PDU that had been received using the normal delivery service to the Subnetwork Interface sublayer

[Note: SIS = Subnetwork Interface Sublayer; CAS = Channel Access Sublayer;
→ = direction of message flow from source to destination sublayer]

B.2.1. **Parameters to C_PHYSICAL_LINK_BREAK**

The C_PHYSICAL_LINK_BREAK primitive has a reason provided by SIS. This reason has the value “Soft Link Terminated for higher priority link” or “Soft Link Terminated to share channel” or “Idle Soft Link Terminated”. This reason is encoded as the Reason value in the PDU sent to the peer to break the link as specified in Section B.7.1.5.

B.2.2. **Errors from C_ Primitives**

The parameters of the C_ service primitives that are clear from the service definition are not listed explicitly. This section lists errors from selected service primitives.

B.2.2.1. **Errors from C_PHYSICAL_LINK REJECTED**

The errors are grouped by where they came from:

1. Rejection by peer with sub-reasons from PHYSICAL LINK REJECTED C_PDU:
    a. Reason Unknown
    b. Broadcast only node
2. DTS rejected transfer of Link Make using D_ EXPEDITED_UNIDATA_REQUEST_REJECTED. The DTS has no peer

mechanism for rejection. The only local mechanism for rejection is TTL expiry. TTL **should** be set to value large enough to ensure this does not happen in practice.

3. Timer expired after final transmission of PHYSICAL LINK REQUEST C_PDU.
4. Local Reasons, defined by this specification:
    a. "Maximum Physical Links In Use".
    b. "ALE Call Timeout"
    c. "ALE Call Rejected"
    d. "Duplex ALE Timeout"
    e. "Link not allowed due to EMCON"
    f. "Physical Link Prevents Broadcast"

These errors are handled as follows:

1. Broadcast only reflects a mis-configuration which needs to be addressed.
2. "Maximum Physical Links In Use" will lead to SIS layer holding off on establishing a physical link to this peer until another physical link has been closed.
3. Other errors will lead to the SIS layer rejecting all S_PDUs queued for the link, and allowing the SIS user to resubmit the messages if it wishes.

B.2.2.2. Errors from C_UNIDATA_REQUEST_REJECTED

C_UNIDATA_REQUEST is an unacknowledged service, and so there are no peer errors. The following errors are noted:

1. DTS rejected transfer D_ UNIDATA_REQUEST_REJECTED. The DTS has no peer mechanism for rejection. The only local mechanism for rejection of a valid PDU is:
    a. TTL expiry.
2. Local reasons, defined in this specification
    a. Broadcast Not Allowed
    b. Address Not Known
    c. No ALE Mapping for Address
    d. Physical Link Needed
    e. Address not Routable

TTL Expiry and the first three local reasons will be passed up to the SIS user along with rejection of the S_PDU. The "Physical Link Needed" error is used when Non-

ARQ data is sent to a peer using ALE, where it is necessary to establish a physical link for all traffic to the peer.

B.3. Interface Primitives Exchanged with the ALE Layer

When ALE is used there is a direct service interaction between the ALE layer and CAS. The primary service elements used are defined in Table B-2. These primitives are used to specify the CAS/ALE interaction. There is no requirement for an implementation to follow this service model.

Table B-2- Nominal Definition of ALE_Primitives for the Interface between the Channel Access sublayer and ALE (non-mandatory, for information-only)

NAME OF PRIMITIVE	DIRECTION	COMMENTS
ALE_LINK_MAKE	CAS →ALE	request to make an ALE link to the node or list of nodes specified
ALE_LINK_COMPLETED	ALE→CAS	report that an ALE link has been made, in response to a local ALE_LINK_MAKE request
ALE_LINK_REJECTED	ALE→CAS	report that a requested ALE link could not be made, typically because of timeout or not available HF frequency
ALE_LINK_REMOTE	ALE→CAS	report that an ALE link has been made, when another node has initiated a link
ALE_LINK_TERMINATE	CAS→ALE	request to terminate the active ALE link
ALE_LINK_TERMINATED	ALE→CAS	report that the active ALE link has been terminated, either by a local operator or by another node in the link
ALE_LINK_RELINK	CAS->ALE	some ALE subsystems will support an option to re-link using ALE Link Management (ALM)

[Note: CAS = Channel Access Sublayer; ALE = ALE Layer
→ = direction of message flow from source to destination sublayer]

This model assumes that ALE searching is controlled by the ALE layer; it is anticipated that the ALE layer will usually search when there is no active link or no link has been requested.

This model does not include sounding, which is discussed in Section B.6.2.7.

B.4. Models of Peer Access

There are two models of channel access, described in the next two sections:

1. Multiple Simultaneous Peer Access

2. Single Peer Access

The model in use **shall** be the same for all nodes using a channel.

For a profile of this standard, the model or models supported need to be specified.

B.5. Multiple Simultaneous Peer Access

The first model is that a node on a channel can simultaneously transmit to multiple peers, utilizing the broadcast nature of the underlying channel. Control of access to the channel is then managed by the MAC layer using CSMA (Annex K) or Wireless Token Ring Protocol (Annex L).

In this model, the Channel Access Sublayer can submit data for all destinations directly to the DTS.

Support of this model is optional. Support of this option by an implementation is indicated with the MULTIPLE ACCESS profile option.

B.6. Single Peer Access

In this second model, the Channel Access Sublayer accesses a channel or channels that can communicate with just one peer. This model **shall** be used in two situations.

1. Where Automatic Link Establishment (ALE) is used to link between peers. When used, this ALE mode of operation **shall** be configured for all nodes accessing the channel; and/or
2. Duplex operation, where exchange of data between two nodes use two exclusive channels, one for transmission and one for reception.

In this model, the Channel Access Sublayer will control the active peer (or active peers with each peer on an independent channel). Requests for transfers to other peers will be rejected, requiring the SIS to queue data and manage traffic for multiple peers where the number of channels available is less than the number of peers for which there is traffic.

This model has four detailed variants set out below. Each of these variants is optional and support is indicated with a profile option.

B.6.1. **Duplex without ALE**

When Duplex configuration is used without ALE, a pair of channels need to be set up between two nodes (one for each direction of transfer). This setup is only viable for one pair of nodes.

Support of this model is optional. Support of this option by an implementation is indicated with the DUPLEX FIXED profile option.

For this configuration, Channel Access operates following the procedures of multiple simultaneous peer access. No queuing in the SIS layer is needed, as there is only one peer. One additional check is made in this configuration:

1. If data is addressed to a unicast STANAG 5066 address other than that of the single duplex peer, it **shall** be rejected with reason code “Address Not Known”

Group addressing **may** be used. Although it is not usually of operational benefit when there is a single peer, it may be helpful for some applications.

B.6.2. **Operation with 1:1 ALE**

A channel may be configured to use ALE (Automatic Link Establishment) to connect to one peer at a time. The procedure defined here can be used with any ALE protocol, including 2G, 3G and 4G. When ALE is used, it **shall** be used for all communication.

Support of this model is optional. Support of this option by an implementation is indicated with the ALE 1:1 profile option.

B.6.2.1. **ALE 1:1 Modes**

1:1 ALE has a choice of two modes:

1. IMPICIT CAS-1. In this mode, a CAS-1 soft link is implicitly established when the ALE link is made and implicitly closed with the ALE link ends. This mode optimizes performance where ALE/CAS-1 links are short and there is usually not a need to relink ALE.
2. EXPLICIT CAS-1. In this mode, a CAS-1 soft link is explicitly negotiated by protocol exchange after the ALE link is established. If the 1:1 ALE mode is supported, this option **shall** be supported. This mode optimizes performance where CAS-1 links are long and commonly relink ALE during the CAS-1 link. It enables a CAS-1 link with a peer to be maintained over an extended period.

The choice of mode is determined by the STANAG 5066 Profile, as defined in the core specification. The mode **shall** be the same for all nodes in a network.

The choice of mode may be influenced by interoperability with Edition 3 systems, as described in Section B.7.2.3.

B.6.2.2. **Queueing Model**

When IMPLICIT CAS-1 is used, there is a single active CAS-1 link. Queueing for this single link and switching between queues is modelled in SIS and specified in Annex A, so that the SIS service controls which link is active.

For EXPLICIT CAS-1, there may be multiple CAS-1 links open and the active link is not visible at the SIS layer. For EXPLICIT CAS-1, the CAS needs to manage an independent queue and DTS state for each active CAS-1 link. The CAS may open and close ALE links as it chooses, associating the correct queue/DTS state with the open ALE link.

B.6.2.3. **Handling Multiple ALE 1:1 Channels**

A node **may** be configured to have multiple channels of this nature, although it is expected that most nodes will be single channel. Where there are multiple channels, the DTS for each channel **shall** operate independently. Each channel in a multi channel system will need a separate modem and ALE unit with an independent DTS. This setup can be considered as a single CAS layer managing multiple modem/ALE/DTS stacks.

B.6.2.4. **Procedure of Operation**

When data arrives for a peer (ARQ or non-ARQ), the following checks are made:

1. If the data is addressed to a broadcast address, it **shall** be rejected with reason code “Physical Link Prevents Broadcast” if a 1:1 ALE link is open. If it is not open, it **may** be handled by the procedure of Multicast ALE set out in Section B.6.3.
2. If data is addressed to an unknown STANAG 5066 address, it **shall** be rejected with reason code “Address Not Known”
3. If an ALE address cannot be determined for the address, it **shall** be rejected with reason code “No ALE Mapping for Address”

When an ALE connection is established to a peer, a CAS-1 link is always established for the peer. In EXPLICIT CAS-1 mode, the CAS-1 soft link establishment procedure described below **shall** be followed, noting that if a CAS-1 link is already open that an ALE link can be established directly. In IMPLICIT CAS-1 mode, the CAS-1 soft link establishment procedure described below **may** be followed, either to determine which edition of STANAG 5066 a peer supports or to negotiate the optional use of short frame sequence number D_PDUs between Edition 4 peers. As an ALE connection initializes the CAS-1 link, the CAS-1 initialization procedure does not need to be used. It is recommended that the CAS-1 link process is only used if necessary.

When closing an ALE link with a peer, the ALE Link Break procedure specified in this Annex **shall** be followed. In EXPLICIT CAS-1 mode, the ALE link **may** be broken with the CAS-1 link remaining open. To close CAS-1 link in EXPLICIT mode, the CAS-1 link break procedure **shall** be used. In IMPLICIT CAS-1 mode, the CAS-1 link break procedure **should not** be used; Break of the CAS-1 link is implicit from ALE Link Break.

If the ALE link is closed externally (e.g., by an operator) the ALE Link Break procedure **shall** be followed, provided that the ALE link is open. In IMPLICIT mode, the CAS-1 link **shall** be treated as closed when the ALE link is closed..

The Channel Access Sublayer **shall** maintain a list of open physical links with maximum number of physical links the same as the number of channels. Each channel will have an independent DTS. When all channels have assigned physical links, any request to open a new physical link **shall** be rejected with error "Maximum Physical Links In Use". Commonly this will be limited to one channel and one active physical link.

When ALE is in use, non-ARQ traffic will need to use an appropriate ALE link. To achieve this in IMPLICIT mode, a physical link is mandated for non-ARQ traffic as well as ARQ traffic. In the event of non-ARQ traffic being submitted when there is not a physical link to the destination, the submission **shall** be rejected with reason "Physical Link Needed".

In EXPLICIT mode, the CAS layer **shall** ensure that an appropriate ALE link is open to carry the non-ARQ traffic.

In normal operation of 1:1 ALE in IMPLICIT mode, the SIS layer will control opening and closing of CAS-1 links to fit with traffic load. When a CAS-1 link is opened, this will lead to an ALE link being established. The CAS-1 Link will be closed cleanly based on Clean Close State specified in Section B.7.2.4. ALE mechanisms are used to open and close the link and CAS-1 link open and close is implicit. This is desirable, as it removes the unnecessary overhead of CAS-1 protocol handshaking.

There are two situations where this basic model of one ALE link per CAS-1 link needs to be extended, which are described in the following two sections.

B.6.2.5. Frequency Change/Validation

For a long running ALE link, a STANAG 5066 application may determine (automatically or with operator input) that operation on a different frequency may be beneficial or choose to validate the current frequency. The approach taken depends on the mode.

For EXPLICIT CAS-1, the current ALE link is either re-linked using Automatic Link Management (ALM) or the link is closed and re-opened.

For IMPICIT CAS-1, the re-linking procedure described in Section B.7.2.7is followed.

B.6.2.6. ALE Link Failure

A second scenario requiring a new ALE link is when an ALE link fails. This will typically be caused by a fading channel that leads to communication being lost between the two end point, both of which will time out the ALE link. The approach taken depends on the mode.

For EXPLICIT CAS-1, a new ALE link is established and the CAS-1 soft link continue.

For IMPICIT CAS-1, the procedure for resuming a link described in Section B.7.2.9 is followed.

B.6.2.7. ALE Sounding

In order to operate efficiently, ALE needs to sound on the frequencies used in order to determine which frequency is best. In a STANAG 5066 and ALE system with light load, it will often be possible and sensible for the ALE component to manage sounding independent of STANAG 5066.

In a highly loaded system, the ALE module may not get sufficient access to the modem to independently perform such sounding. In such a setup, it will often be important for the STANAG 5066 system to pro-actively manage and co-ordinate ALE sounding. The details of achieving this are outside the scope of this specification.

B.6.3. Operation with Multicast ALE (non-ARQ)

It is also possible to use ALE to open a link to multiple nodes. This **may** be done when multicast traffic is processed, for example in support of a multicast protocol such as ACP 142. This link is used for non-ARQ traffic only.

NOTE: While it is technically possible to manage soft links over multicast ALE, the process for managing this appears unduly complex, so this is not supported in this edition of AComP 5066.

Support of this model is optional. Support of this option by an implementation is indicated with the MULTICAST ALE profile option.

Multicast ALE is initiated when non-ARQ data is sent to a STANAG 5066 broadcast address. The ALE call addresses used will be determined from the STANAG 5066 broadcast address and an appropriate ALE Group Call initiated.

In order to use Multicast ALE, any 1:1 ALE links need to be closed first. The SIS layer has knowledge of this requirement and will close down any soft links and associated ALE 1:1 links before using the Multicast ALE service.

Multinode ALE calls need to be closed by the call initiator. This may be done after an idle period or when the SIS layer requests establishment of a soft link and associated 1:1 ALE link.

B.6.4. **Duplex with 1:1 ALE**

In order to use Duplex transfer between a pair of nodes without ALE, it is necessary to configure a fixed frequency for each direction of transfer between a pair of nodes.

Support of this model is optional. Support of this option by an implementation is indicated with the DUPLEX WITH ALE profile option.

When a pair of nodes each have duplex support and ALE is used, duplex communication can be negotiated. In order to do this, a node shall determine if the peer can support duplex communication. There are two methods to achieve this:

1. A priori knowledge; or
2. The Data Transfer Sublayer enables determination that a peer node is able to support duplex by use of EOWs.

If a node determines that both peers support duplex, duplex communication **may** be negotiated using ALE. It is recommended that this is done. On receiving an ALE link, a node with a duplex peer **may** use ALE to negotiate a second connection with the peer. If this is successfully done, the first ALE link **shall** be used by the node which initiated it to send data and the second ALE link **shall** be used to send data in the reverse direction.

B.7. Channel Access Protocol Type 1 and C_PDUs

Where ALE is not used, the local node will be connected to a channel, which can be fixed frequency or with variable frequency determined by an external process (e.g., selection of frequency based on a fixed schedule). Other nodes may be connected to the same channel.

The local node **may** know the addresses of all other nodes that can be connected to the channel. If this is the case and data is addressed to another node, it **shall** be rejected with reason code “Address Not Known”.

The co-ordination of the making and breaking of Physical Links (hereinafter referred to as 'CAS 1 Linking Protocol') between two nodes **shall** be performed solely by the Channel Access Sublayer. If ALE is not used, the CAS 1 linking protocol **shall** be used.

If the CAS 1 Linking Protocol is omitted (when ALE is used) and the response to ARQ-data DPDUs is a Warning DPDU indicating that a connection is not made, then the CAS 1 Linking Protocol **shall** be followed and the data resent.

If a slave node (receiving station) accepts an ALE link and a CAS-1 Physical-Link Request C_PDU is received, then the receiving node **shall** respond in accordance with the requirements of the 'CAS 1 Linking Protocol', accepting or rejecting the request as is appropriate for its state.

When ALE is used, the CAS 1 linking protocol **shall** respond to the reception of a type 4 PHYSICAL_LINK_BREAK C_PDU by sending a type 5 PHYSICAL_LINK_BREAK_CONFIRM C_PDU as specified in Section B.4.2.2. The ALE link is then closed and the physical link **shall** be broken at that point.

A Node shall use a Physical Link to support control and data exchange for Soft-Link Data Exchange Sessions as requested by the Subnetwork Interface Sublayer.

A Node **may** have Physical Links with more than one other node at a time, up to one Physical Link per remote node; i.e., a Node may “Make” a new Physical Link with another node before it “Breaks” any Physical links.

There **shall** be no explicit peer-to-peer communication required to switch from use of one Physical Link to another.

When 1:1 ALE is used, the Channel Access sublayer enforces that there is at most one active ALE link per channel and that for each active ALE link there is a single physical link associated with that ALE link.

The Channel Access Sublayer CAS 1 linking protocol **shall** communicate with peer sublayers in other nodes using the protocols defined here in order to:

1. Make and break physical links

2. Deliver S_PDUs between Subnetwork Interface Sublayers at the local node and remote node(s).

B.7.1. Type 1 Channel Access Sublayer Data Protocol Units (C_PDUs)

The following C_PDUs **shall** be used for peer-to-peer communication between Channel Access Sublayers in the local and remote node(s):

C_PDU NAME	*Type Code*
DATA C_PDU	TYPE 0
PHYSICAL LINK REQUEST	TYPE 1
PHYSICAL LINK ACCEPTED	TYPE 2
PHYSICAL LINK REJECTED	TYPE 3
PHYSICAL LINK BREAK	TYPE 4
PHYSICAL LINK BREAK CONFIRM	TYPE 5
PHYSICAL LINK BREAK REQUEST	TYPE 6
PHYSICAL LINK RESUME	TYPE 7
PHYSICAL LINK RESUME	TYPE 8
PHYSICAL LINK RESUME REJECT	TYPE 9
ALE RELINK	TYPE

The first argument and encoded field of all C_PDUs **shall** be the C_PDU Type

The remaining format and content of these C_PDUs **shall** be as specified in the subsections that follow.

Unless noted otherwise, argument values encoded in the C_PDU bit-fields **shall** be mapped into the fields in accordance with CCITT V.42, 8.1.2.3, i.e.:

1. when a field is contained within a single octet, the lowest bit number of the field **shall** represent the lowest-order (i.e., least-significant-bit) value;
2. when a field spans more than one octet, the order of bit values within each octet **shall** decrease progressively as the octet number increases. The lowest bit number associated with the field **shall** represent the lowest-order (i.e., least significant bit) value.

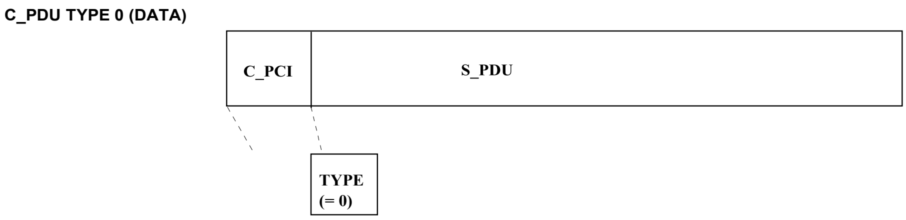
Unless noted otherwise, bit-fields specified as NOT USED **shall** be encoded with the value ‘0’ (i.e., zero).

B.7.1.1. DATA C_PDU

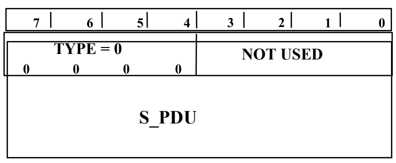
Type :

“0” = DATA C_PDU

Encoding :



(a) Generic Encoding



(b) Bit-Field Map

Figure B-1: Generic Encoding and Bit-Field Map of the DATA C_PDU

Description :

The DATA (TYPE 0) C_PDU **shall** be used to send an encapsulated S_PDU from the local node to a remote node.

The *Type* argument **shall** be encoded in the first four-bit field of the DATA C_PDU as shown in Figure B-1. The value of the *Type* argument for the DATA C_PDU **shall** be zero.

The remaining octets of the DATA C_PDU **shall** contain the encapsulated S_PDU and only the encapsulated S_PDU.

For the Channel Access sublayer request to the lower layers of the subnetwork to deliver a C_PDU, the delivery service requirements for a DATA C_PDU **shall** be the same as the S_PDU that it contains, i.e.:

a) C_PDUs **shall** be sent using the normal Data Delivery service provided by the lower sublayer;
b) the DELIVERY mode specified by the Subnetwork Interface Sublayer for the encapsulated S_PDU (i.e., ARQ, non-ARQ, etc.) also **shall** be assigned to the C_PDU by the Channel Access sublayer as the delivery mode to be provided by the lower sublayer.

B.7.1.2. PHYSICAL LINK REQUEST C_PDU

Type :

"1" = PHYSICAL LINK REQUEST

Encoding :

	MSB 7	6	5	4	3	2	1	LSB 0
0	Type=1				Not Used	Short	Ed4	LINK

Figure B-2: Generic Encoding and Bit-Field Map of the PHYSICAL LINK REQUEST C_PDU

Description :

The PHYSICAL LINK REQUEST C_PDU **shall** be transmitted by a Channel Access sublayer to request the making of the Physical Link.

The PHYSICAL LINK REQUEST C_PDU **shall** consist of the arguments *Type, Link*, *Short*, and *Ed4*
The value of the *Type* argument for the PHYSICAL LINK REQUEST C_PDU **shall** be ‘1’ (i.e., one), encoded as a four-bit field as shown in Figure B-2.

The bits not used in the encoding of the PHYSICAL LINK REQUEST C_PDU **shall** be reserved for future use and not used by any implementation.

The Ed4 bit indicates to the peer that the local node supports edition 4 (or subsequent version) of the protocol. If the peer is known to support edition 4 (or subsequent) this bit **shall** be set. If the peer is known to not support edition 4 (or subsequent) this bit **shall** not be set. Otherwise this bit **may** be set in order to discover if the peer supports edition 4.

If the Ed4 bit is set, the Short bit **may** be set. This indicates a request to use D_PDUs with short LFSN, which can give performance advantage at very low speed.

The LINK bit was used in Edition 3 to request an exclusive link. This capability is not supported by default in Edition 4. If the HARD LINKS deprecated optional profile is supported, then it **shall** be used as specified in Section B.9.1. Otherwise, this bit shall be set to zero on transmission. Error Code 5 (Exclusive Links Not Supported) shall be used in response to a request with this bit set.

When a PHYSICAL LINK REQUEST C_PDU is transmitted, the local Node is not linked to another Node and therefore the ARQ transmission mode is not supportable by the Data Transfer Sublayer.
Therefore, the PHYSICAL LINK REQUEST C_PDU **shall** be sent by the Channel Access Sublayer requesting the lower-layer Expedited Non-ARQ Data Service (i.e., transmission using Type 8 D_PDUs) in accordance with the Annex C specification of the Data Transfer Sublayer.

A Channel Access sublayer which receives a PHYSICAL LINK REQUEST C_PDU **shall** respond with either a PHYSICAL LINK ACCEPTED (TYPE 2) C_PDU or a PHYSICAL LINK REJECTED (TYPE 3) C_PDU, as appropriate.

B.7.1.3. PHYSICAL LINK ACCEPTED C_PDU

Type :

"2" = PHYSICAL LINK ACCEPTED

Encoding :

	MSB 7	6	5	4	3	2	1	LSB 0
0	Type=2				Not Used	Short	Ed4	Not Used

Figure B-3: Generic Encoding and Bit-Field Map of the PHYSICAL LINK ACCEPTED C_PDU

Description :

The PHYSICAL LINK ACCEPTED (TYPE 2) C_PDU **shall** be transmitted by a peer sublayer as a positive response to the reception of a TYPE 1 C_PDU (PHYSICAL LINK REQUEST).

PHYSICAL LINK ACCEPTED (TYPE 2) C_PDU **shall** consist only of the argument *Type*. [Note: A Node can never request the establishment of more than one Physical Link with another given node, therefore, the request for which this message is a response is uniquely identified by the node address of the node to which the PHYSICAL LINK ACCEPTED C_PDU is sent.]

The *Type* argument **shall** be encoded as a four-bit field containing the binary value 'two' as shown in Figure B-3.

The bits not used in the encoding of the PHYSICAL LINK ACCEPTED C_PDU **shall** be reserved for future use and not used by any implementation. They **shall** be set to zero on transmission.

The Ed4 bit indicates to the peer that the local node support edition 4 (or subsequent version) of the protocol. If the equivalent bit was set in the PHYSICAL LINK REQUEST C_PDU which is being responded to this bit **shall** be set. Otherwise, this bit **shall not** be set.

If the Ed4 bit is set and the Short bit was set in the PHYSICAL LINK REQUEST C_PDU which is being responded to, the Short bit **may** be set. This indicates that D_PDUs with short LFSN, which can give performance advantage at very low speed **shall** be used with this link in both directions. If Ed4 bit is set and the Short bit is not set, D_PDUs with short LFSN **shall not** be used. This means that between Ed4 peers, short LFSN is only used if both peers agree.

When a PHYSICAL LINK ACCEPTED (TYPE 2) C_PDU is transmitted the local Node is not linked to another Node and therefore the ARQ transmission mode is not supportable by the Data Transfer Sublayer. Therefore, the PHYSICAL LINK ACCEPTED C_PDU **shall** be sent by the Channel Access Sublayer requesting the lower-layer's Expedited Non-ARQ Data Service (i.e., transmission using Type 8 D_PDUs) in accordance with the Annex C specification of the Data Transfer Sublayer.

B.7.1.4. PHYSICAL LINK REJECTED C_PDU

Type :

"3" = PHYSICAL LINK REJECTED

Encoding :

7	6	5	4	3	2	1	0
TYPE = 3_{10}				REASON MSB	--	--	LSB

Figure B-4: Generic Encoding and Bit-Field Map of the PHYSICAL LINK REJECTED C_PDU

Description :

The PHYSICAL LINK REJECTED (TYPE 3) C_PDU **shall** be transmitted by a peer sublayer as a negative response to the reception of a TYPE 1 C_PDU (PHYSICAL LINK REQUEST).

The PHYSICAL LINK REJECTED (TYPE 3) C_PDU **shall** consist of two arguments: *Type* and *Reason.*

The *Type* argument **shall** be encoded as a four-bit field containing the binary value 'three' as shown in Figure B-4.

The *Reason* argument **shall** be encoded in accordance with Figure B-4 and the following table:

Reason	Value
Reason Unknown	0
Broadcast-Only-Node	1
Higher-Priority-Link-Request-Pending	2
Not Used	3
Too many active links	4
Exclusive Links Not	5
unspecified	6-15

The value 0, indicating an unknown reason for rejecting the Make request for the physical link, is always a valid reason for rejection. Reasons corresponding to values in the range 6-15 are currently unspecified and unused in this standard.

When a PHYSICAL LINK REJECTED C_PDU is transmitted the local Node is not linked to another Node and therefore the ARQ transmission mode is not supportable by the Data Transfer

Note that “Link busy” is not included in the list of possible reasons for rejecting anode’s request to establish a link since this function is provided by the WARNING frame type of the Data Transfer Sublayer, as specified in Annex C.

The PHYSICAL LINK REJECTED C_PDU **shall** be sent by the Channel Access Sublayer requesting the lower-layer Expedited Non-ARQ Data Service (i.e., transmission using Type 8 D_PDUs) in accordance with the Annex C specification of the Data Transfer Sublayer.

B.7.1.5. **PHYSICAL LINK BREAK C_PDU**

Type :
“4” = PHYSICAL LINK BREAK
Encoding :

	7	6	5	4	3	2	1	0
0	TYPE = 4				REASON			
	0	1	0	0	MSB	--	--	LSB

Figure B-5: Generic Encoding and Bit-Field Map of the PHYSICAL LINK BREAK C_PDU

Description :
The PHYSICAL LINK BREAK C_PDU **shall** be transmitted by either of the peer Channel Access sublayers involved in an active Physical Link to request the breaking of the link.

The PHYSICAL LINK BREAK C_PDU **shall** consists of two arguments: *Type and Reason.*

The *Type* argument **shall** be encoded as a four-bit field containing the binary value ‘four’ as shown in Figure B-5.

The *Reason* argument **shall** be encoded in accordance with Figure B-5 and the following table:

Reason	Value
Reason Unknown	0
Higher-Layer-	1
Switching to Broadcast- Data-	2
Switching link to send higher priority data	3
No More Data	4
Switching link for	5
Operator request	6
unspecified	7-15

The value 0, indicating an unknown reason for requesting the breaking of a physical link, is always a valid reason. Reasons corresponding to values in the range 5-15 currently are not defined in this standard.

A peer sublayer which receives the PHYSICAL LINK BREAK C_PDU **shall** immediately declare the Physical Link as broken and respond with a PHYSICAL LINK BREAK (TYPE 5) C_PDU as specified in section B.3.1.6 below.

The PHYSICAL LINK BREAK C_PDU **shall** be sent by the Channel Access Sublayer requesting the lower-layer Expedited Non-ARQ Data Service (i.e., transmission using Type 8 D_PDUs) in accordance with the Annex C specification of the Data Transfer Sublayer.
[Note: The reason the PHYSICAL LINK BREAK C_PDU is sent with the non-ARQ data service, even though a physical link exists at the time that it is sent, is because the receiving peer will immediately declare the Link as broken. The receiving peer will therefore not have time to send ARQ acknowledgements for the C_PDU.]

B.7.1.6. PHYSICAL LINK BREAK CONFIRM C_PDU

Type :

“5” = PHYSICAL LINK BREAK CONFIRM

Encoding :

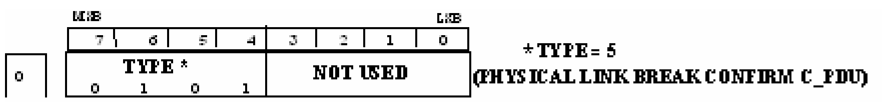


Figure B-6: Generic Encoding and Bit-Field Map of the PHYSICAL LINK BREAK CONFIRM C_PDU

Description :

The PHYSICAL LINK BREAK CONFIRM (TYPE 5) C_PDU **shall** be transmitted by a Channel Access sublayer as a response to a TYPE 4 “PHYSICAL LINK BREAK” C_PDU.

The PHYSICAL LINK BREAK CONFIRM (TYPE 5) C_PDU **shall** consist of the single argument *Type*.

The *Type* argument **shall** be encoded as a four-bit field containing the binary value ‘five’ as shown in Figure B-6.

The PHYSICAL LINK BREAK CONFIRM (TYPE 5) C_PDU **shall** be sent by the Channel Access Sublayer requesting the lower-layer Expedited Non-ARQ Data Service (i.e., transmission using Type 8 D_PDUs) in accordance with the Annex C specification of the Data Transfer Sublayer.

Upon receiving a PHYSICAL LINK BREAK CONFIRM (TYPE 5) C_PDU, the peer which initiated the breaking of the Link **shall** declare the Link as broken.

B.7.1.7. PHYSICAL LINK BREAK REQUEST C_PDU

Type :

"6" = PHYSICAL LINK BREAK REQUEST

Encoding :

	MSB 7	6	5	4	3	2	1	LSB 0
0	Type=6				Not Used	Reason		
1	Not Used				Priority			

Figure B-7: Encoding of the PHYSICAL LINK BREAK REQUEST C_PDU

Description :

The PHYSICAL LINK BREAK REQUEST (TYPE 6) C_PDU **may** be transmitted by a peer sublayer to indicate to the peer a desire to break the link. This enables a link to be closed cleanly, with co-operation between both ends resulting in the clean close state specified in Section B.7.2.4.

This C_PDU is defined in Edition 4 and **shall not** be transmitted to Edition 3 peers.

This C_PDU is used to request termination of ALE link and associated physical link.

The three bit Reason field indicates the reason that the break is being requested. MSB of Reason is bit 2 and LSB is bit 0. Values for Reason are specified in the following table:

Reason	**Value**	**Description**
No Data	000	The node sending PHYSICAL LINK BREAK REQUEST has no more data to send and suggests the link is terminated
Higher Priority	001	There is data for another destination of higher priority than traffic for the current link and link needs to be closed to transmit it.
Channel Share	010	Link needs to be closed so that the channel can be shared fairly with another link.
Operator Request	011	An operator has requested that the link be broken
Not used	100 - 111	Reserved for future versions of this protocol.

The Priority field **shall** be set to the priority of the higher priority traffic when Reason is

"Higher Priority". For other Reasons, the priority field **shall** be set to 0.

When a PHYSICAL LINK BREAK REQUEST (TYPE 6) C_PDU is transmitted the local Node is linked to the peer Node. Therefore, the PHYSICAL LINK BREAK REQUEST C_PDU **shall** be sent by the Channel Access Sublayer requesting the lower-layer's ARQ Data Services in accordance with the Annex C specification of the Data Transfer Sublayer. This will mean that the PHYSICAL LINK BREAK REQUEST C_PDU will arrive after any data sent in the same transmission. Before the peer breaks the link, it will need to ensure that all data prior to the PHYSICAL LINK BREAK REQUEST C_PDU has been received, which can be done by examination of the DTS ARQ window.

B.7.1.8. PHYSICAL LINK RESUME C_PDU

Type :
"7" = PHYSICAL LINK RESUME

Encoding :

	MSB 7	6	5	4	3	2	1	LSB 0
0	Type=7				Not Used			

Figure B-8: Encoding of the PHYSICAL LINK RESUME C_PDU

Description :
The PHYSICAL LINK RESUME (TYPE 7) C_PDU **shall** be transmitted by a peer sublayer to indicate to the peer that it is desired to resume a physical link.

This C_PDU is defined in Edition 4 and **shall not** be transmitted to Edition 3 peers.

B.7.1.9. PHYSICAL LINK RESUME ACCEPT C_PDU

Type :
"8" = PHYSICAL LINK RESUME ACCEPT

Encoding :

	MSB 7	6	5	4	3	2	1	LSB 0
0	Type=8				Not Used			

Figure B-9: Encoding of the PHYSICAL LINK RESUME ACCEPT C_PDU

Description :
The PHYSICAL LINK RESUME ACCEPT (TYPE 8) C_PDU **shall** be transmitted

by a peer sublayer to indicate to the peer that it is agreed to resume a physical link.

This C_PDU is defined in Edition 4 and **shall not** be transmitted to Edition 3 peers.

B.7.1.10. **PHYSICAL LINK RESUME REJECT C_PDU**

Type :
"9" = PHYSICAL LINK RESUME REJECT

Encoding :

	MSB 7	6	5	4	3	2	1	LSB 0
0	Type=9				Not Used			

Figure B-10: Encoding of the PHYSICAL LINK RESUME REJECT C_PDU

Description :
The PHYSICAL LINK RESUME REJECT (TYPE 9) C_PDU **shall** be transmitted by a peer sublayer to indicate to the peer that it will not resume a physical link.

This C_PDU is defined in Edition 4 and **shall not** be transmitted to Edition 3 peers.

B.7.1.11. **ALE RELINK C_PDU**

Type :
"10" = ALE RELINK

Encoding :

	MSB 7	6	5	4	3	2	1	LSB 0
0	Type=10				Not Used	Reason		

Figure B-11: Encoding of the ALE RELINK C_PDU

Description :
The ALE RELINK (TYPE 10) C_PDU **may** be transmitted by a peer sublayer to indicate to the peer a desire to break an ALE link, while retaining the physical link.

This C_PDU is defined in Edition 4 and **shall not** be transmitted to Edition 3 peers.

The three bit Reason field indicates the reason that the relink is being requested. MSB of Reason is bit 2 and LSB is bit 0. Values for Reason are specified in the following table:

Reason	Value	Description
ALE Quality	000	The current link is determined to be poorer quality than expected. Link needs to be closed so that ALE can see if a better channel is available.
ALE Time	001	ALE link has been running for a long time. Link needs to be closed so that ALE can see if a better channel is available.
Not used	010 - 111	Reserved for future versions of this protocol.

When a ALE RELINK (TYPE 10) C_PDU is transmitted the local Node is linked to the peer Node. Therefore, the ALE RELINK C_PDU **shall** be sent by the Channel Access Sublayer requesting one of the lower-layer's Expedited ARQ Data Services in accordance with the Annex C specification of the Data Transfer Sublayer. This well enable relinking to be performed as early as possible.

B.7.2. **Type 1 Channel Access Sublayer Peer-to-Peer Communication Protocol**

Requirements on the Type 1 Channel Access Sublayer Peer-to-Peer Communication Protocols are specified in this section and the subsections below.

The Channel Access sublayer **shall** perform all peer-to-peer communications for protocol control using the following C_PDUs:

C_PDUs for Peer-to-Peer Protocol	*Type Code*
PHYSICAL LINK REQUEST	TYPE 1
PHYSICAL LINK ACCEPTED	TYPE 2
PHYSICAL LINK REJECTED	TYPE 3
PHYSICAL LINK BREAK	TYPE 4
PHYSICAL LINK BREAK CONFIRM	TYPE 5
PHYSICAL LINK BREAK REQUEST	TYPE 6
PHYSICAL LINK RESUME	TYPE 7
PHYSICAL LINK RESUME ACCEPT	TYPE 8
PHYSICAL LINK RESUME REJECT	TYPE 9
ALE RELINK	TYPE 10

All peer-to-peer communications **shall** be done by the Channel Access sublayer requesting the lower-layer Expedited Non-ARQ Data Service (i.e., transmission using Type 8 D_PDUs in accordance with the Annex C specification of the Data Transfer Sublayer).

The criteria for accepting or rejecting Physical Links Requests **shall** be as follows. A request to establish a Physical Link **shall** be accepted in Single Peer Access mode, as receipt of the request implies that the 1:1 underlying channel is open. In Multiple Simultaneous Peer Access mode, the Physical Link **shall** be accepted, unless the number of active physical links exceeds a configurable limit or another operational reason.

B.7.2.1. Protocol for Making a Physical Link

In the specification that follows of the protocol for making a physical link, the node which requests the physical link is referred to as the Caller or Calling node, while the node which receives the request will be referred to as the Called node.

The protocol for making the physical link **shall** consist of the following steps:

<u>Step 1-Caller:</u>

Initiation of the link is in response to C_PHYSICAL_LINK_MAKE request from SIS,

If ALE is being used, the following initial caller procedure **shall** be used:

a) If ALE is being used, an ALE link to the peer **shall** be requested using ALE_LINK_MAKE.
b) If the ALE link is rejected or timed out with ALE_LINK_REJECT, the request for physical link **shall** be rejected with reason "ALE Call Timeout" or "ALE Link Rejected". If ALE_LINK_COMPLETE is received, the link is made.
c) If Duplex ALE is being used, wait for the incoming ALE call indicated by ALE_LINK_REMOTE. If this does not arrive within a configurable time limit the first ALE call **shall** be released using ALE_LINK_TERMINATE and the request for physical link **shall** be rejected with reason "Duplex ALE Timeout".
d) If the Ed4 support status of the peer is not known or if it is desired to use short frame sequence number D_PDUs to an Ed4 peer, the second stage of this procedure **shall** be followed. Otherwise in IMPLICIT CAS-1 mode, C_PHYSICAL_LINK_MADE is sent to the SIS, D_CONNECTION_MADE is sent to the DTS, and the link is made.

The following procedure is followed when ALE is not being used, or in EXPLICIT CAS-1 mode, or when the initial ALE procedure requires full CAS-1. This procedure is referenced above as "second stage".

a) The Calling Node's Channel Access Sublayer shall send a PHYSICAL LINK REQUEST (TYPE 1) C_PDU to initiate the protocol,
b) Upon sending the PHYSICAL LINK REQUEST (TYPE 1) C_PDU the Channel Access Sublayer shall start a timer which is set to a value

greater than or equal to the maximum time required by the Called Node to send its response (this time depends on the modem parameters, number of re-transmissions, etc.),

c) The maximum time to wait for a response to a PHYSICAL LINK REQUEST (TYPE 1) C_PDU shall be a configurable parameter in the implementation of the protocol.

Step 2-Called:

If ALE is being used, the called node waits for an incoming ALE link and receipt of ALE_LINK_COMPLETE from ALE to CAS. If duplex ALE is being used for the peer, the second ALE call is initiated using ALE_LINK_MAKE. If this second call fails, the initial ALE call is cleared using ALE LINK TERMINATE. On ALE completion D_CONNECTION_MADE is signaled to the DTS for each connected ALE peer.

Then an inbound C_PDU is waited for. C_PDUs will be received as a data indications from the DTS, noting that DATA C_PDUs will arrive as D_UNIDATA_INDICATION and other C_PDUs will arrive as D_EXPEDITED_UNIDATA_INDICATION.

- If it is a DATA C_PDU or a PHYSICAL LINK ACCEPTED C_PDU, the link is indicated made by:
    - SIS is informed that the CAS-1 soft link is open use C_PHYSICAL_LINK_MADE; and
    - The DATA C_PDU is passed to SIS.
    - NOTE: Receipt of PHYSICAL LINK ACCEPTED C_PDU is not expected. In the event that it is received, it is seen as preferable to accept the link rather than to clear it.
- If it is a PHYSICAL LINK REQUEST C_PDU, the procedure described below for handling PHYSICAL LINK REQUEST C_PDU is followed. If it is any other C_PDU, the ALE link is cleared.

If no C_PDU arrives within a configurable timer, the ALE link **shall** be cleared using ALE_LINK_TERMINATE.

The following procedure is followed when ALE is not used or when a PHYSICAL LINK REQUEST C_PDU arrives over an ALE link.

a) On receiving a PHYSICAL LINK REQUEST (TYPE 1) C_PDU, a Called node **shall**
   determine whether or not it can accept or reject the request as follows:
   (1) If the Called node has the maximum number of active Physical Links, the Called node **shall** reject the request with reason “Too many active links”,
   (2) otherwise, the Called node **shall** accept the request for a Physical Link.

b) After determining if it can accept or reject the PHYSICAL LINK REQUEST, a Called node
   shall respond as follows:

(1) if a Called node accepts the physical link request, it **shall** respond with a PHYSICAL LINK ACCEPTED (TYPE 2) C_PDU,
(2) otherwise, when a Called node rejects the physical link request, it **shall** respond with a PHYSICAL LINK REJECTED (TYPE 3) C_PDU.

b) After a PHYSICAL LINK ACCEPTED (TYPE 2) C_PDU is sent, the called channel access sublayer **shall** declare the physical link made and transition to a state in which it executes the protocol for data exchange using C_PDUs.
c) If further PHYSICAL LINK REQUEST (TYPE 1) C_PDUs are received from the same address after the link is made, the Channel Access sublayer **shall** again reply with a PHYSICAL LINK ACCEPTED (TYPE 2) C_PDU.
d) If at least one DATA (TYPE 0) C_PDU is not received on a newly activated Physical Link after waiting for a specified maximum period of time, the Called Node **shall** abort the Physical Link and declare it inactive.
e) The maximum period of time to wait for the first DATA (TYPE 0) C_PDU before aborting a newly activated Physical Link **shall** be a configurable parameter in the implementation.

When a physical link is accepted, SIS is informed with C_PHYSICAL_LINK_MADE. If ALE is not being used the DTS is informed with D_CONNECTION_MADE (this step is done earlier when ALE is used).

<u>Step 3. Caller</u>

The caller follows this procedure when the second procedure in step 1 was followed.

There are two possible outcomes to the protocol for making a physical link: success or failure:

a) Upon receiving a PHYSICAL LINK ACCEPTED (TYPE 2) C_PDU, the calling Channel Access Sublayer **shall** proceed as follows:
   (1) the Calling node **shall** declare the Physical Link as *successfully Made*,
   (2) send C_PHYSICAL_LINK_MADE to the SIS and send D_CONNECTION_MADE to the DTS; otherwise,
b) upon receiving a PHYSICAL LINK REJECTED (TYPE 3) C_PDU, the Channel Access Sublayer **shall** declare the Physical Link as *Failed,* and send C_PHYSICAL_LINK_REJECTED to the SIS. otherwise,
c) upon expiration of its timer without any response having been received from the remote node, the Channel Access Sublayer **shall** repeat Step 1 (i.e. send a PHYSICAL LINK REQUEST (TYPE 1) C_PDU and set a response time) and await again a response from the remote node.

The maximum number of times the Caller sends the PHYSICAL LINK REQUEST (TYPE 1) C_PDU without a response from the called node of either kind **shall** be a configurable parameter in the implementation of the protocol.

After having repeated Step 1 the configurable maximum number of times without any response having been received from the remote node, the Caller's Channel Access Sublayer **shall** declare the protocol to make the physical link as *Failed*, and send C_PHYSICAL_LINK_REJECTED to the SIS.

Figure B-12 and Figure B-13 show the nominal procedures followed by the Caller and Called Channel Access Sublayers for Making a Physical Link, when ALE is not used or after ALE initialization and CAS-1 is used. This standard acknowledges that other implementations may meet the stated requirements.

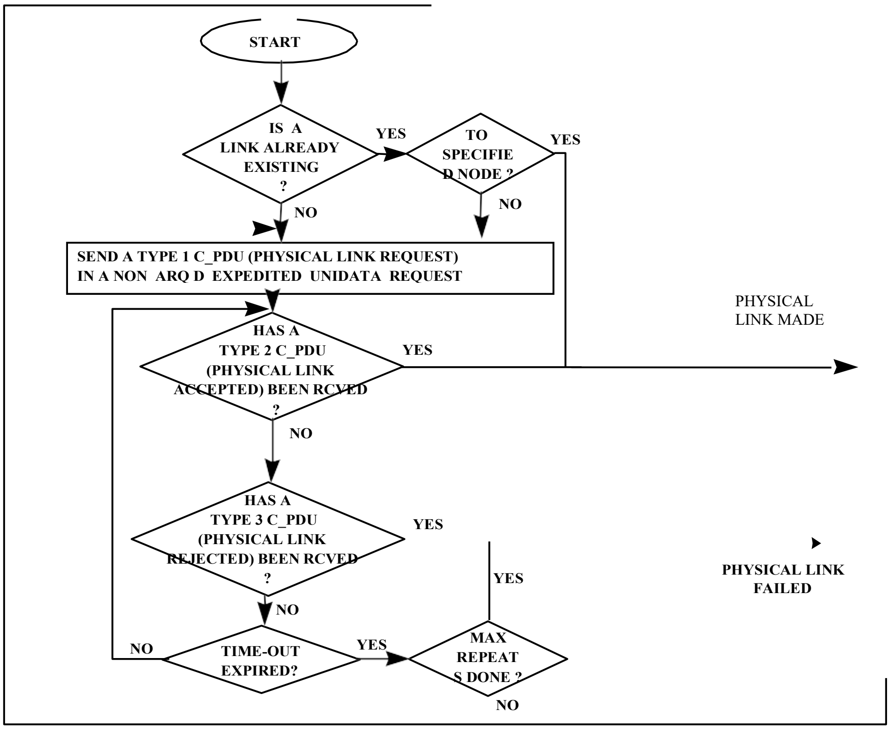


Figure B-12: Type 1 Channel Access Sublayer protocol for Making a Physical Link (Caller Peer)

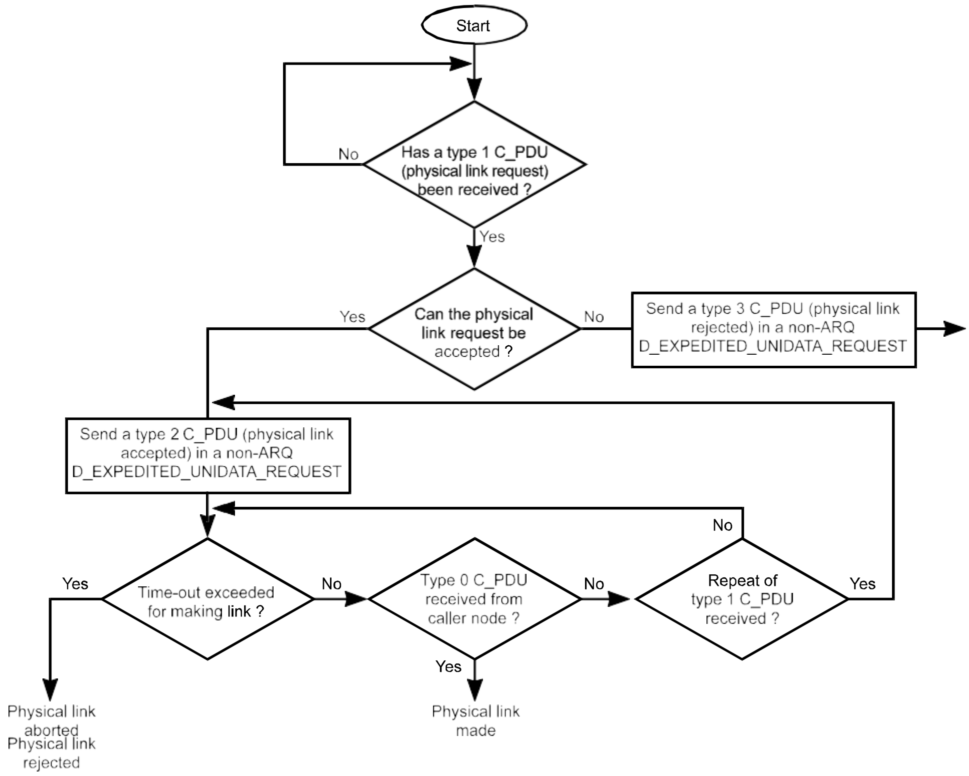


Figure B-13: Type 1 Channel Access Sublayer Protocol for Making a Physical Link (Called Peer)

After having declared the Physical Link as successfully made, the peer Channel Access Sublayers do not carry out any further handshake in order to confirm the successful operation of the Link.

B.7.2.2. Determining Edition 4 Support

The physical link setup is also used to determine support for this edition of the protocol (or subsequent editions). On receipt of a PHYSICAL LINK REQUEST C_PDU with the Ed4 bit set, a node can determine that a peer supports edition 4.

It is likely that an edition 3 (or earlier) implementation receiving a PHYSICAL LINK

REQUEST C_PDU with the Ed4 bit set will ignore this bit and respond with a PHYSICAL LINK ACCEPT C_PDU which does not have the Ed4 bit set. This will enable the local node to determine that the peer does not support edition 4. This mechanism provides a robust way to determine if a peer supports edition 4 (or subsequent).

It is possible, but unlikely, that the edition 3 peer will respond to a C_PDU with the Ed4 bit set by treating it as an error and responding with a WARNING D_PDU. In the event that this happens the local peer **shall** record that the peer does not support edition 4 and **shall** repeat the link establishment without the Ed4 bit set. For subsequent links, the record of edition support can avoid repeating the failed link attempt.

When both peers support edition 4 (or subsequent) the default DTS behavior will be to use the family of D_PDUs with 16 bit Frame Sequence Number. Use of the Short bit allows peers to negotiate use of D_PDUs with 8 bit Frame Sequence Number. This could be beneficial in scenarios where it is known that speeds will be 1200 bps or less, as the D_PDU header overhead is slightly reduced.

B.7.2.3. **Interoperability with Edition 3 Systems using ALE 1:1**

Edition 3 does not specify how to use ALE 1:1, but a number of vendors have implemented a capability. This specification defines use of ALE 1:1 with Ed3, as it will only use Ed3 PDUs to interoperate with and Ed3 system.

The choice of mode (IMPLICIT CAS-1 vs EXPLICIT CAS-1) should be made based on how the Ed3 system works. It is known that there are Ed3 systems which used both modes, although it is believed that EXPLICIT CAS-1 is more common.

B.7.2.4. **Clean Close State**

Prior to terminating an ALE Link or a physical link, it is desirable that the CAS 1 state is “clean”, with no pending traffic in either direction. This maximizes efficiency and minimizes errors.

A node is in a clean state when:

a. It has no data sent by the peer to acknowledge; and
b. It has no data sent that has not been acknowledged

A node will naturally reach this state when a link goes idle and there is no data left to be transferred. A model of “close links when there is nothing to transfer” can be provided cleanly.

NOTE: A node can only determine if it is in clean state when it is the nodes turn to transmit, typically immediately after it has received data from the peer.

Clean Close State is only needed when a link is closed where a different link may be opened after the link is closed. If an ALE link is being closed for re-linking with expectation of resuming the current link, there is no need to reach Clean Close State, as the procedure set out in this section B.7.2.9 can be followed to ensure no loss.

In order to support requests to close busy links, a mechanism to get links into a clean state is provided. When a request to close a link is received, the procedure described in this section **shall** be used to reach a clean close state with peers that support this version of the protocol. This procedure **shall not** be used with Edition 3 peers.

If the node is not in a clean close state, the requirements are to finish transfer of active traffic on the link and to cease using the link for new traffic. In order to reach clean close state, a PHYSICAL LINK BREAK REQUEST C_PDU is sent, which contains the reason for closing the link. This message is sent using ARQ and so does not need to be timed or repeated.

Once a node has sent a PHYSICAL LINK BREAK REQUEST C_PDU it will monitor link state and **shall** initiate closing the link as soon as the node can determine that it is in clean close state, which will usually be after receiving data from the peer.

When a node receives a PHYSICAL_LINK_BREAK_REQUEST, its actions will depend on the Reason provided.

Where the reason is “Channel Share”, or “Operator Request”, the node **shall** take actions to move the link to clean close state. No new Data C_PDUs **shall** start transmission apart from client confirmations, noting that this control may need interaction with the DTS. Where a Data C_PDU has started transmission (i.e., some D_PDUs have been sent) then the rest of the C_PDU **shall** be transmitted. Note that reaching clean close state may take a number of transmissions in each direction. The node **shall** monitor link state and **shall** initiate closing the link as soon as the node reaches clean close state.

The “No Data” reason is advisory. If the node receiving this has data to transmit, it **shall** continue to use the link. The node **shall** monitor link state and **shall** initiate closing the link as soon as the node reaches clean close state.

Action on the “Higher Priority” reason depends on the priority. New C_PDUs of lower priority **shall not** start transmission. However, new C__PDUs of the same or higher priority **shall** start transmission. The node **shall** monitor link state and **shall** initiate closing the link as soon as the node reaches clean close state.

B.7.2.5. Protocol for Terminating an ALE Link (IMPLICIT CAS-1)

When ALE is used in IMPLICIT CAS-1 mode, the following procedure **shall** be used for terminating the ALE link. The procedure also closes all physical links associated with the ALE link.

When ALE is used, one of the nodes will terminate the link using ALE_LINK_TERMINATE and the ALE layer will communicate termination to other nodes, where the CAS layer will receive ALE_LINK_TERMINATED.

ALE links should generally be terminated promptly, either when there is no data to send or when SIS wishes to schedule transfer of data to a different peer which will require a new ALE link and a new physical link.

If C_PHYSICAL LINK BREAK is received from SIS, the Clean Close Procedure specified in Section B.7.2.4 **shall** be followed using the reason provided by the C_PHYSICAL_LINK BREAK.

Then the ALE Link Break procedure is started, which will terminate the ALE link for all nodes.

Similarly, if ALE analysis determines that the link should be broken (due either to link quality being poorer than anticipated or to the length of time), the Clean Close Procedure specified in Section B.7.2.4 **shall** be followed using the appropriate ALE reason. Then the ALE Link Break procedure is started, which will terminate the ALE link for all nodes.

Otherwise the ALE Link Break procedure shall be started when any of the following conditions apply:

1. For each physical link, the node is in Clean Close State as described in Section B.7.2.4 or DTS has reported link failure with D_CONNECTION_LOST; and
2. If any Non-ARQ data has been sent or received over the link, that a configurable timer since last non-ARQ data was sent or received has expired; and
3. If the ALE link is multi-node, that no traffic between other nodes has been detected more recently than a configurable time in the past.

In addition to this there **shall** be a configurable ALE Link timer. If no data has been received over the ALE link (or detected between other nodes in a multi-node link) within this timer, then the ALE Link Break procedure **shall** be followed.

The ALE Link Break procedure is for a node to use ALE_LINK_TERMINATE whenever the preceding conditions are met. Any physical links associated with the ALE link **shall** be declared broken, by sending D_CONNECTION_TERMINATED to the DTS.

If the link was in Clean Close state on termination, then C_PHYSICAL_LINK_BROKEN **shall** be sent to the SIS.

If the link was non in Clean Close state, which may be due to DTS initiated Link Closure or ALE timeout, then C_PHYSICAL_LINK_BROKEN **may** be sent the SIS. Alternatively, another ALE link **may** be initiated following the protocol for resuming a physical link, as specified in Section B.7.2.9.

If an ALE_LINK_TERMINATED is received at any time, the ALE link has been terminated by a peer or a local operator. A peer is expected only to terminate an ALE link when in Clean Close state. Any physical links associated with the ALE link **shall** be declared broken, by sending D_CONNECTION_TERMINATED to the DTS and C_PHYSICAL_LINK_BROKEN to the SIS.

B.7.2.6. Protocol for Terminating an ALE Link (EXPLICIT CAS-1)

When ALE is used in EXPLICIT CAS-1 mode, the following procedure **shall** be used for terminating the ALE link.

The procedure specified in Section B.7.2.8is followed to break the CAS-1 link.

After the CAS-1 link is closed, the ALE link is terminated by the initiator when it receives PHYSICAL LINK BREAK CONFIRM. The ALE link is closed using ALE_LINK_TERMINATE.

B.7.2.7. Protocol for ALE Relinking

There are situations where it is desired to continue a physical link, but to relink ALE. This might be because the performance of the current link is too poor or because the current link has been running for an extended period, and it is desired to determine if a better link can be achieved.

In EXPLIT CAS-1 mode, ALE may be relinked at any time by either node. This can be done using ALE_LINK_RELINK, or using ALE_LINK_TERMINATE immediately followed by ALE_LINK_MAKE.

The following procedure is used in IMPLICIT CAS-1 mode.

This procedure **shall not** be used with Edition 3 peers.

When a node determines a desire to relink, it **shall** send a ALE RELINK C_PDU with value of either “ALE Quality” or “ALE Time” dependent on reason for relinking. The node **shall** then wait for the DTS acknowledgement of the ALE RELINK C_PDU, to ensure that the peer is aware that the relink procedure is being initiated. The node **shall** then immediately relink at ALE level. If the ALE supports ALE Link Management (ALM), the link **shall** be relinked using ALE_LINK_RELINK. Otherwise the current link is terminated using ALE_LINK_TERMINATE and then a new link made immediately using ALE_LINK_MAKE.

Following relinking, the node performing the relinking **shall** initiate the state resumption procedure specified in Section B.7.2.9.

On receiving an ALE RELINK C_PDU, a node **shall** send DTS acknowledgement of the C_PDU and **shall not** send any other data. It will then wait for the peer to continue the relinking procedure. This node **may** be notified of relinking using ALM. Alternatively it may receive ALE_LINK_TERMINATED. In the latter case, it **shall not** initiate any new ALE requests for a configurable period, but **shall** wait for an incoming ALE_LINK_REMOTE. If the incoming request is not from the relinking peer and there is a mechanism to reject the incoming call, the node **may** do so.

B.7.2.8. Protocol for Breaking a Physical Link

In the specification that follows of the protocol for breaking a physical link, the node which requests the breaking of the Physical Link will be referred to as the Initiator or initiating node, while the node which receives the request will be referred to as the Responder or responding node. This protocol is used when ALE is not used or in EXPLICIT CAS-1 mode.

Physical Links will be broken in the following situations:

1. If C_PHYSICAL LINK BREAK is received from SIS; or
2. D_CONNECTION_LOST reported by the DTS;.

If D_CONNECTION_LOST is reported by the DTS, the SIS is informed with C_PHYSICAL_LINK_BROKEN.

For C_PHYSICAL_LINK BREAK, the Clean Close Procedure specified in Section B.7.2.4shall be followed using the reason provided by the C_PHYSICAL_LINK BREAK.

It is recommended that SIS client flow control is used to avoid large queue build up in the DTS. However, there needs to be some measure of queueing in DTS in order for the DTS to operate efficiently. When a link is broken prior to all traffic being sent, it is generally desirable to close the link quickly. For this reason, link break procedure makes use of expedited data, which will ensure that the link break protocol is not blocked behind other traffic on the link.

The protocol for breaking the Physical Link **shall** consist of the following steps:

Step 1: Initiator

a) To start the protocol, the Initiator's Channel Access Sublayer **shall** send a type 4 C_PDU (PHYSICAL LINK BREAK).
b) Upon sending this C_PDU the Channel Access Sublayer **shall** start a timer which is set to a value greater than or equal to the maximum time required by the Called Node to send its response (this time depends on the modem parameters, number of re-transmissions, etc.).

Step 2: Responder

a) Upon receiving the type 4 C_PDU the Responder's Channel Access Sublayer **shall** declare the Physical link as *broken* and send a PHYSICAL LINK BREAK CONFIRM (TYPE 5) C_PDU. Then D_CONNECTION_TERMINATED is sent to the DTS and C_PHYSICAL_LINK_BROKEN is sent to the SIS.

Step 3: Initiator

a) Upon receiving a PHYSICAL LINK BREAK CONFIRM (TYPE 5) C_PDU, the Initiator's Channel Access Sublayer **shall** declare the Physical Link as *broken by sending D_CONNECTION_TERMINATED to the DTS and C_PHYSICAL_LINK_BROKEN to the SIS.*.

b) Upon expiration of its timer without any response having been received from the remote node, the Initiator's Channel Access Sublayer **shall** repeat step 1 and wait again for a response from the remote node.

c) After having repeated Step 1 a maximum number of times (left as a configuration parameter) without any response having been received from the remote node, the Initiator's Channel Access Sublayer **shall** declare the Physical Link as *broken by sending D_CONNECTION_TERMINATED to the DTS and C_PHYSICAL_LINK_BROKEN to the SIS.*.

Figure B-14(a) and (b) show the procedures followed by the Initiator and Responding Channel Access Sublayers for Breaking a Physical Link.

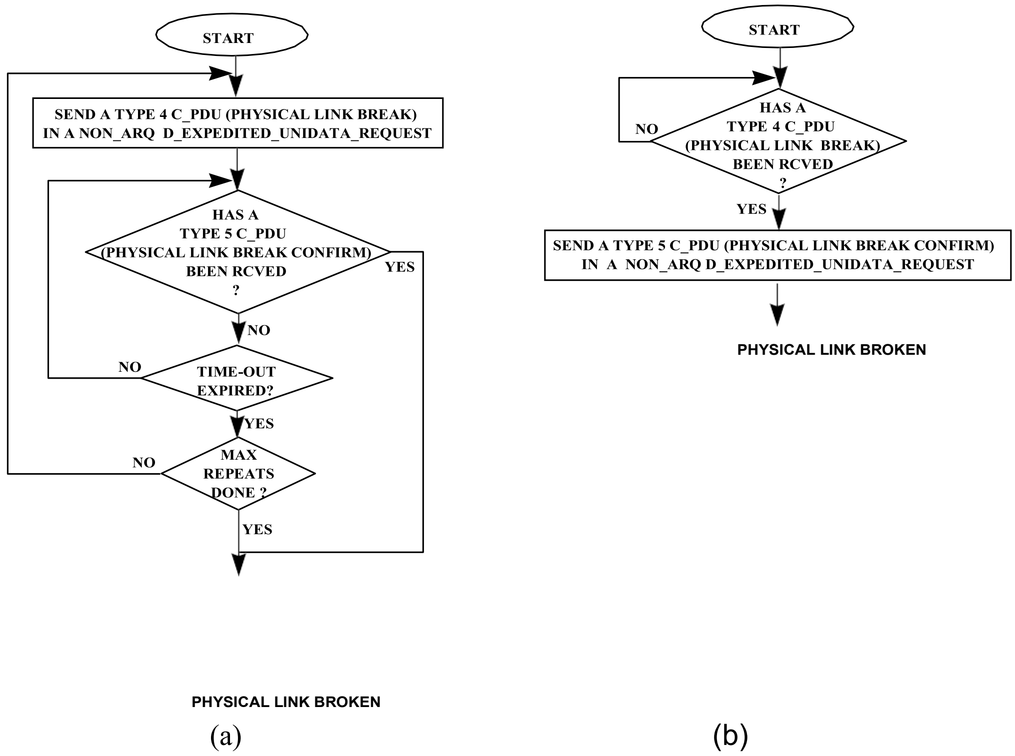


Figure B-14: Type 1 Channel Access Sublayer protocol for Breaking a Physical Link
(a) Initiator Peer. (b) Responding Peer.

B.7.2.9. Protocol for Resuming a Physical Link

The process **may** be used with Edition 4 peers when an ALE link has failed. In this situation, it is likely that both peers will be aware that the ALE link has failed. This process allows both peers to continue without reset of CAS-1 status. This process **shall not** be used with Edition 3 peers.

For this process to work, both nodes need to have preserved CAS-1 / ARQ status from the previous ALE link.

The node initiating this process shall send a PHYSICAL LINK RESUME C_PDU using the Expedited ARQ service and wait for a response.

A node receiving a PHYSICAL LINK RESUME C_PDU **may** accept the resumption if it has preserved CAS-1 state since the previous ALE link, which it does by sending a PHYSICAL LINK RESUME ACCEPT C_PDU using the Expedited ARQ service. It then continues operation with standard protocol, and waits for the ALE Link initiator to transmit data on the ALE link.

The receiving node **may** also reject the request by sending a PHYSICAL LINK RESUME REJECT C_PDU using the Expedited ARQ service. The receiver will then proceed as for reception of a new ALE link,

When the initiator receives a PHYSICAL LINK RESUME ACCEPT C_PDU, it will then proceed with data transfer following normal procedures starting with the preserved CAS-1 state.

When the initiator receives a PHYSICAL LINK RESUME REJECT C_PDU, it will then reset local CAS-1 and ARQ status, and the proceed as for a newly initiated ALE link.

In the event that a node which has preserved CAS-1 status receives an ALE link and the first data received is not a PHYSICAL LINK RESUME PDU, the ARQ status is reset.

B.8. Protocol for Exchanging Data C_PDUs.

The protocol for exchanging C_PDUs between peers is simple and straightforward, since the Channel Access Sublayer relies entirely on its supporting (i.e, Data Transfer) sublayer to provide delivery services.

The sending peer **shall** accept S_PDUs from the Subnetwork Interface Sublayer, envelop them in a “DATA” C_PDU (by adding the C_PCI) and send them to its receiving peer via its interface to the Data Transfer Sublayer.

The receiving peer **shall** receive DATA C_PDUs from the Data Transfer Sublayer interface, check them for validity, strip off the C_PCI, and deliver the enveloped S_PDU to the Subnetwork Interface Sublayer.

B.9. Deprecated Service

The Hard Links service specified in Edition 3 are optional in this edition and deprecated. It is anticipated that this services will be removed in future updates of this specification. This are retained to ensure interoperability with Edition 3 systems, although it is believed that such use is minimal.

Specification of these services is primarily by reference to the text in Edition 3, which ensures full alignment.

B.9.1. **Hard Links**

This optional deprecated service is defined as the HARD LINKS profile option.

Hard Links are specified in Edition 3 Annex A in a manner that does not conflict with any of the Edition 4 specification. So use of the Edition 3 specification is straightforward. Section B.3.2 specifies how to establish a physical link which may either be a hard link or a soft link.

B.10. Changes in Edition 4

This annex has the following changes relative to Edition 3.

1. Explicit setting out of single peers access vs multiple peer access as part of the channel model. This was implicit in Edition 3. It needs to be made explicit in order to clearly document ALE and Duplex operation.

2. Use of 1:1 ALE for connecting to a peer is documented explicitly.

3. The model of duplex operation is set out, both with and without ALE.

4. Where ALE is not used, the specification requires CAS-1 linking. The generic text around handling ALE-like systems is removed.

5. The CAS-1 negotiation is extended to determine if the peer supports Edition 4, and if so to negotiate the choice of D_PDUs to use on the soft link.

6. Some new Error types are introduced, arising from ALE and Duplex support.

7. C_CHANNEL_AVAILABILITY service primitive is removed. It is listed, but not defined. It does not appear to be useful.

8. Errors for C_ primitives explicitly listed.

9. Support for hard links is optional and deprecated.

10. Incorporation of notes from STANAG 5066 Edition 3 Annex H.

11. Removal of handling Expedited Data is optional and deprecated, in line with Annex A.

12. Add ALE Link Termination Procedure

13. Add support for ALM and Relinking

14. Added EXPLICIT CAS-1 and IMPLICIT CAS-1 modes