

DIN EN 61158-4-2

ICS 35.100.20; 35.160

Ersatz für
DIN EN 61158-4-2:2013-01
Siehe Anwendungsbeginn

**Industrielle Kommunikationsnetze –
Feldbusse –
Teil 4-2: Protokollspezifikation des Data Link Layer
(Sicherheitsschicht) –
Typ 2-Elemente
(IEC 61158-4-2:2014);
Englische Fassung EN 61158-4-2:2014, nur auf CD-ROM**

Industrial communication networks –
Fieldbus specifications –
Part 4-2: Data-link layer protocol specification –
Type 2 elements
(IEC 61158-4-2:2014);
English version EN 61158-4-2:2014, only on CD-ROM

Réseaux de communication industriels –
Spécifications des bus de terrain –
Partie 4-2: Spécification du protocole de la couche liaison de données –
Éléments de type 2
(CEI 61158-4-2:2014);
Version anglaise EN 61158-4-2:2014, seulement en CD-ROM

Gesamtumfang 340 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2014-09-19 angenommene Europäische Norm als DIN-Norm ist 2015-09-01.

Für DIN EN 61158-4-2:2013-01 besteht eine Übergangsfrist bis 2017-09-19.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN 61158-400:2013-08.

Für dieses Dokument ist das nationale Arbeitsgremium K 956 „Feldbus“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Die enthaltene IEC-Publikation wurde vom SC 65C „Industrial networks“ erarbeitet.

Das IEC-Komitee hat entschieden, dass der Inhalt dieser Publikation bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu dieser Publikation angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees die Publikation

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf eine Norm ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils neueste gültige Ausgabe der in Bezug genommenen Norm.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe der Norm.

Der Zusammenhang der zitierten Normen mit den entsprechenden Deutschen Normen ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation. Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für die vorliegende Norm erfüllt

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN 61158-4-2 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt, und Abschnitt 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für viele der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen

Einführung

Ein Feldbus ist ein digitaler serieller Datenbus für die Kommunikation zwischen Geräten der industriellen Automatisierungstechnik wie beispielsweise Messeinrichtungen, Stellgliedern, Reglern und speicherprogrammierbaren Steuerungen. Die Normen der Reihe DIN EN 61158 legen eine Anzahl von Feldbus-Protokolltypen fest. Die Reihe DIN EN 61158-4 legt das Protokoll des Data Link Layer für die einzelnen Typen fest.

Änderungen

Gegenüber DIN EN 61158-4-2:2013-01 wurden folgende Änderungen vorgenommen:

- a) ControlNet-Objekte in 7.2 auf den neuesten Stand gebracht;
- b) das TCP/IP-Interface-Objekt in 7.5 erweitert;
- c) das Ethernet Link-Objekt in 7.6 erweitert;
- d) das CCO-Objekt in 7.8 erweitert;
- e) das DLR-Objekt in 7.9 auf den neuesten Stand gebracht;
- f) das QoS-Objekt in 7.10 auf den neuesten Stand gebracht;
- g) ein Port-Objekt in 7.11 hinzugefügt;
- h) die DL-Zustandsmaschinen in 8.1 und 9.2 auf den neuesten Stand gebracht;
- i) das DLR-Protokoll in Abschnitt 10 auf den neuesten Stand gebracht;
- j) redaktionelle Überarbeitung;
- k) Indikator-Verhalten in A.2.2 und A.2.3 auf den neusten Stand gebracht.

Frühere Ausgaben

DIN V 19245-1: 1988-01
 DIN 19245-1: 1991-04
 DIN 19245-2: 1991-04
 DIN EN 50170/1: 1997-07
 DIN EN 50170/2: 1997-07
 DIN EN 50170/3: 1997-07
 DIN EN 50254: 1999-07
 DIN EN 50170/A2: 2000-08
 DIN EN 50170/A1: 2002-11
 DIN EN 50170/A3: 2002-11
 DIN EN 61158-4: 2005-05
 DIN EN 61158-4-2: 2008-09, 2013-01

Nationaler Anhang NA (informativ)

1 Anwendungsbereich^{N1)}

1.1 Allgemeines

Der Data Link Layer stellt grundlegende zeitkritische Kommunikation für den Datentransfer zwischen Geräten in einer Automatisierungsumgebung bereit.

Dieses Protokoll stellt Kommunikationsmöglichkeiten sequenziell und in einer zyklischen synchronen Weise für alle teilnehmenden Data Link-Einheiten bereit. Im Vordergrund geplanter Zugriff ist für zeitkritische Aktivitäten zusammen mit im Hintergrund ungeplantem Zugriff für weniger kritische Aktivitäten vorhanden.

Deterministische und synchronisierte Transfers können bei zyklischen Intervallen bis zu 1 ms und bei Geräte-Entfernungen von 25 km bereitgestellt werden. Diese Leistung ist dynamisch und online durch Neukonfigurierung der Parameter der lokalen Verbindung veränderbar während des normalen Betriebs. Durch ähnliche Mittel können DL-Verbindungen und neue Geräte während des Betriebs hinzugefügt oder entfernt werden.

Dieses Protokoll liefert Mittel, um die Uhrzeitsynchronisierung entlang einer erweiterten Verbindung mit einer besseren Genauigkeit als 10 µs beizubehalten.

Dieses Protokoll optimiert jede Zugriffsmöglichkeit durch Verkettung mehrerer DLSDUs und dazugehöriger DLPCI in eine einzelne DLPDU, dadurch verbessert sich die Datentransfer-Effizienz für Data Link-Einheiten, welche aktiv mehrere Datenströme beziehen.

Die maximale Systemgröße ist eine unbegrenzte Anzahl von Verbindungen mit 99 Knoten, jeder mit 255 DLSAP-Adressen. Jede Verbindung hat maximal 224 ähnliche Peer- und Publisher-DLCEPs.

^{N1)} Die Nummerierung aus IEC 61158-4-2 wurde beibehalten.

1.2 Spezifikationen

Diese Norm legt Folgendes fest:

- a) Prozeduren für zeitgerechten Datentransfer und Informationssteuerung von einer Data Link-Benutzereinheit zu einer gleichartigen Benutzereinheit sowie zwischen der Data Link-Einheit, welche den verteilten Data Link-Dienst-Anbieter bildet;
- b) die Struktur der Feldbus-DLPDUs, welche verwendet werden für den Transfer von Daten und Informationssteuerung durch das Protokoll dieser Norm und ihre Darstellung als Physical Layer.

1.3 Prozeduren

Die Prozeduren sind definiert in Form von

- a) den Interaktionen zwischen gleichartigen DL-Einheiten (DLEs) durch den Austausch der Feldbus-DLPDUs,
- b) den Interaktionen zwischen einem DL-Dienst-(DLS-)Anbieter und einem DLS-Benutzer im selben System durch den Austausch von DLS-Primitiven,
- c) den Interaktionen zwischen einem DLS-Anbieter und einem Ph-Dienst-Anbieter im selben System durch den Austausch von Ph-Dienstprimitiven.

1.4 Anwendbarkeit

Diese Prozeduren sind auf Instanzen der Kommunikation zwischen Systemen anwendbar, welche die zeitkritischen Kommunikationsdienste innerhalb des Data Link Layer des OSI oder des Feldbusreferenzmodells unterstützen und die Fähigkeit erfordern, in einer offenen Systemverbundumgebung zu verbinden.

Profile stellen ein einfaches Multiattribut Hilfsmittel zur Zusammenfassung der Implementierungsfähigkeiten bereit und unterstützen dadurch die Anwendbarkeit für unterschiedliche zeitkritische Kommunikationsanforderungen.

1.5 Konformität

Diese Norm legt auch Konformitätsanforderungen für Systeme fest, die diese Prozeduren implementieren. Sie enthält jedoch keine Prüfungen, um die Konformität mit diesen Anforderungen nachzuweisen.

3.3 Allgemeine Begriffe

Für die Anwendung dieses Dokuments gelten die folgenden Begriffe.

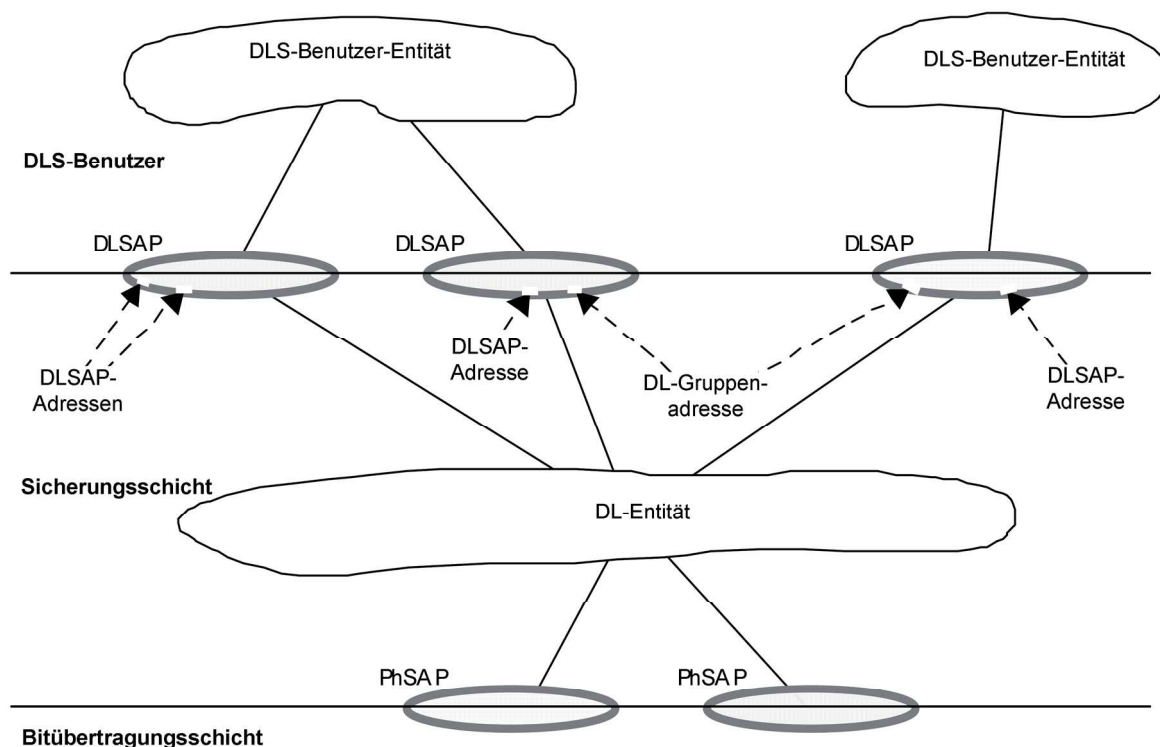
ANMERKUNG Viele Definitionen gelten für mehr als einen Protokolltyp; sie werden nicht zwangsläufig von allen Protokolltypen verwendet.

3.3.1

DL-Segment, Verbindungszweig, lokaler Verbindungszweig

(en: DL-segment, link, local link)

einzelnes DL-Subnetzwerk, in dem alle angeschlossenen DLEs direkt, d. h. ohne zwischengeschaltete DL-Weitergabe, kommunizieren können, solange alle an einem Kommunikationsvorgang teilnehmenden DLEs während der Dauer(n) der Kommunikationsversuche gleichzeitig auf das DL-Subnetzwerk reagieren können



ANMERKUNG 1 DLSAPs und PhSAPs werden als Ellipsen dargestellt, die sich über die Grenze zweier benachbarter Schichten erstrecken.

ANMERKUNG 2 DL-Adressen werden als gekennzeichnete, kleine Lücken (Zugangspunkte) im DLL-Teil eines DLSAP dargestellt.

ANMERKUNG 3 Eine einzelne DL-Entität kann mehrere DLSAP-Adressen und DL-Gruppenadressen, die einem einzelnen DLSAP zugeordnet sind, besitzen.

Bild 1 – Beziehungen von DLSAPs, DLSAP-Adressen und DL-Gruppenadressen

3.3.2 DLSAP

(en: DLSAP)

eindeutiger Punkt, an dem eine einzelne DL-Entität einer einzelnen Entität auf höherer Schicht DL-Dienste bereitstellt

Anmerkung 1 zum Begriff: Diese Definition ist abgeleitet von ISO/IEC 7498-1 und wurde hier zum besseren Verständnis für die kritische Unterscheidung zwischen DLSAPs und ihren DL-Adressen wiederholt. (Siehe Bild 1.)

3.3.3 DL(SAP)-Adresse

(en: DL(SAP)-address)

entweder eine individuelle DLSAP-Adresse, die einen einzelnen DLSAP eines einzelnen DLS-Benutzers kennzeichnet, oder eine DL-Gruppenadresse, die möglicherweise mehrere DLSAPs jeweils eines einzelnen DLS-Benutzers kennzeichnet

Anmerkung 1 zum Begriff: Diese Terminologie wird gewählt, da ISO/IEC 7498-3 die Verwendung der Benennung „DLSAP-Adresse“ zur Bezeichnung von mehr als einem einzelnen DLSAP bei einem einzelnen DLS-Benutzer nicht zulässt.

3.3.4

(individuelle) DLSAP-Adresse

(en: (individual) DLSAP-address)

DL-Adresse, die nur einen einzigen DLSAP innerhalb des erweiterten Verbindungszweigs kennzeichnet

Anmerkung 1 zum Begriff: Eine einzelne DL-Entität kann mehrere DLSAP-Adressen, die einem einzelnen DLSAP zugeordnet sind, besitzen.

3.3.5

erweiterter Verbindungszweig

(en: extended link)

DL-Subnetzwerk, das aus der maximalen Menge von Verbindungszweigen besteht, die über DL-Relais-Entitäten geschaltet werden können, wobei ein einziger DL-Namensraum (DL-Adressraum) gemeinsam genutzt wird; in diesem Subnetzwerk können alle angeschlossenen DL-Entitäten miteinander entweder direkt oder über eine oder mehrere der zwischengeschalteten DL-Relais-Entitäten kommunizieren

Anmerkung 1 zum Begriff: Ein erweiterter Verbindungszweig besteht möglicherweise aus nur einem einzelnen Verbindungszweig.

3.3.6

Rahmen

(en: frame)

umgangssprachliches Synonym für DLPDU

3.3.7

DL-Gruppenadresse

(en: group DL-address)

DL-Adresse, die möglicherweise mehr als einen DLSAP innerhalb des erweiterten Verbindungszweigs kennzeichnet. Eine einzelne DL-Entität kann mehrere DL-Gruppenadressen, die einem einzelnen DLSAP zugeordnet sind, besitzen. Eine einzelne DL-Entität kann auch eine einzelne DL-Gruppenadresse, die mehr als einem DLSAP zugeordnet ist, besitzen

3.3.8

Knoten

(en: node)

einzelne DL-Entität, wie sie auf einem lokalen Verbindungszweig existiert

3.3.9

empfangender DLS-Benutzer

(en: receiving DLS-user)

Benutzer der DL-Dienste, der als Empfänger von DL-Benutzerdaten agiert

Anmerkung 1 zum Begriff: Ein Benutzer der DL-Dienste kann gleichzeitig sendender und empfangender DLS-Benutzer sein.

3.3.10

sendender DLS-Benutzer

(en: sending DLS-user)

Benutzer der DL-Dienste, der als Quelle von DL-Benutzerdaten agiert

NA.3.4 Zusätzliche Begriffe für Typ 2

3.4.1

tatsächliches Paketintervall

API

(en: actual packet interval, API)

Maß, wie häufig eine spezifische Verbindung ihre L-Paketdaten erzeugt

3.4.2

zuweisen

(en: allocate)

eine Ressource aus einem allgemeinen Bereich der ausschließlichen Nutzung durch eine spezifische Entität zuordnen

3.4.3

Anwendung

(en: application)

Funktion oder Datenstruktur, für die Daten konsumiert oder produziert werden

3.4.4

Anwendungsobjekte

(en: application objects)

mehrere Objektklassen, die den Nachrichtenaustausch über das Netzwerk und innerhalb des Gerätes zur Laufzeit verwalten und bereitstellen

3.4.5

Attribut

(en: attribute)

Beschreibung eines von außen sichtbaren Merkmals oder einer von außen sichtbaren Funktion eines Objekts

Anmerkung 1 zum Begriff: Die Attribute eines Objekts enthalten Informationen zu veränderlichen Teilen eines Objekts. Sie liefern üblicherweise Statusinformationen oder steuern den Betrieb eines Objekts. Attribute können auch das Verhalten eines Objekts beeinflussen. Attribute werden in Klassenattribute und Instanzattribute unterteilt.

3.4.6

Verhalten

(en: behavior)

Anzeige dafür, wie das Objekt auf bestimmte Ereignisse reagiert

Anmerkung 1 zum Begriff: Seine Beschreibung schließt die Beziehung zwischen Attributwerten und Diensten ein.

3.4.7

Bit

(en: bit)

Informationseinheit, die aus einer 0 oder einer 1 besteht

Anmerkung 1 zum Begriff: Ein Bit ist die kleinste Dateneinheit, die übertragen werden kann.

3.4.8

Austastung oder Austastzeit

(en: blanking or blanking time)

erforderliche Dauer nach einer Übermittlung, bevor ein Knoten wieder empfangen darf

3.4.9

Client

(en: client)

- 1) Objekt, das die Dienste eines anderen (Server-)Objekts zur Ausführung einer Aufgabe nutzt;
- 2) Initiator einer Nachricht, auf die ein Server reagiert

3.4.10

Kommunikationsobjekte

(en: communication objects)

Komponenten, die den Austausch von Nachrichten über das Netzwerk zur Laufzeit verwalten und bereitstellen, z. B. das ControlNet-Objekt, der UCCM (en: Unconnected Message Manager)

3.4.11

Verbindung

(en: connection)

logische Verknüpfung zwischen zwei Anwendungsobjekten innerhalb eines Gerätes oder in unterschiedlichen Geräten

3.4.12

Verbindungs-ID

CID

(en: connection ID, CID)

Kenntnis, die einer Übertragung im Zusammenhang mit einer bestimmten Verbindung zwischen Produzenten und Konsumenten zugewiesen ist und die einen bestimmten Teil von Anwendungsinformationen kennzeichnet

Anmerkung 1 zum Begriff: Auf der Data Link Layer entspricht diese Kennung einem generischen Tag.

3.4.13

zyklische Redundanzprüfung

CRC

(en: cyclic redundancy check, CRC)

Restwert, der aus einem Feld von Daten errechnet und als repräsentative Signatur für das Feld verwendet wird

3.4.14

zyklisch

(en: cyclic)

Benennung zur Beschreibung von Ereignissen, die sich regelmäßig wiederholen

3.4.15

Taubheit

(en: deafness)

Situation, in welcher der Knoten nicht die Moderator-DLPDU, aber anderen Verkehr auf dem Verbindungszweig hören kann

3.4.16

Gerät

(en: device)

physische Hardware-Verbindung zum Verbindungszweig

Anmerkung 1 zum Begriff: Ein Gerät kann mehr als einen Knoten umfassen.

3.4.17

DLPDU

(en: data-link protocol data unit, DLPDU)

Protokolldateneinheit auf der Data Link Layer

Anmerkung 1 zum Begriff: Eine DLPDU besteht aus einer MAC-ID, null oder mehr L-Paketen und einer Blockprüfzeichenfolge, wie sie von einer zugehörigen PhE übertragen oder empfangen wurde.

3.4.18

Endezeichen

(en: end delimiter)

eindeutiger Satz von M_Zeichen, der das Ende einer DLPDU angibt

3.4.19

Fehler

(en: error)

Nichtübereinstimmung zwischen einem berechneten, beobachteten oder gemessenen Wert oder Zustand und dem festgelegten oder theoretisch richtigen Wert oder Zustand

3.4.20**erweiterter Verbindungszweig**

(en: extended link)

Verbindungszweige, die über DL-Router miteinander verbunden sind (manchmal auch als lokales Netzwerk bezeichnet)

3.4.21**FCS-Fehler**

(en: FCS error)

Fehler, der auftritt, wenn der berechnete Wert der Blockprüfzeichenfolge nach Empfang aller Oktette in einer DLPDU nicht mit dem erwarteten Rest übereinstimmt

3.4.22**festes Tag**

(en: fixed tag)

Bezeichner (Tag) aus zwei Oktetten, der einen bestimmten Dienst kennzeichnet, welcher ausgeführt werden soll entweder

- a) von jenem empfangenden Knoten auf dem lokalen Verbindungszweig, der eine festgelegte MAC-ID aufweist, oder
- b) von allen empfangenden Knoten auf dem lokalen Verbindungszweig

Anmerkung 1 zum Begriff: Die Identifikation des/der Zielknoten(s) ist in dem zwei Oktette langen Tag enthalten.

3.4.23**Rahmen**

(en: frame)

umgangssprachliches Synonym für DLPDU

3.4.24**Blockprüfzeichenfolge****BPF**

(en: Frame Check Sequence, FCS)

redundante Daten, die mittels Hash-Funktion aus einem Block von Daten innerhalb einer DLPDU (Rahmen) abgeleitet und zusammen mit diesem Datenblock gespeichert oder übertragen werden, um Datenverfälschungen zu erkennen

3.4.25**generisches Tag**

(en: generic tag)

Bezeichner (Tag) aus drei Oktetten, der einen bestimmten Teil von Anwendungsinformationen kennzeichnet

3.4.26**Sicherheitsband**

(en: guardband)

Zeitschlitz für die Übertragung der Moderator-DLPDU

3.4.27**implizites Token**

(en: implicit token)

Mechanismus, der das Recht zur Übertragung steuert

Anmerkung 1 zum Begriff: Über das Medium wird keine eigentliche Token-Nachricht übertragen. Jeder Knoten verfolgt die MAC-ID des Knotens, von dem er annimmt, dass dieser derzeit das Recht zur Übertragung hat. Das Recht zur Übertragung wird von Knoten zu Knoten durch Aufzeichnen des Knotens, der zuletzt übertragen hat, weitergegeben. Damit ein fehlender Knoten im Durchlauf übersprungen werden kann, wird eine Schlitzzeit verwendet.

3.4.28

Speicherzelle für implizites Token

(en: implicit token register)

Speicherzelle, welche die MAC-ID des Knotens, der das Recht zur Übertragung hat, enthält

3.4.29

Instanz

(en: instance)

tatsächliches physisches Vorkommen eines Objekts innerhalb einer Klasse, das eines von mehreren Objekten innerhalb derselben Objektklasse identifiziert

BEISPIEL „Kalifornien“ ist eine Instanz der Objektklasse „Staat“.

Anmerkung 1 zum Begriff: Die Benennungen „Objekt“, „Instanz“ und „Objektinstanz“ beziehen sich auf eine spezifische Instanz.

3.4.30

Instanzattribut

(en: instance attribute)

Attribut, das für eine Objektinstanz eindeutig ist und sich nicht auf die Objektklasse bezieht

3.4.31

instanciiert

(en: instantiated)

Objekt, das in einem Gerät erzeugt wurde

3.4.32

Keeper

(en: Keeper)

Objekt, das für die Verteilung von Konfigurationsdaten des Verbindungszweigs an alle Knoten auf dem Verbindungszweig verantwortlich ist

3.4.33

Verbindungszweig

(en: link)

Zusammenstellung von Knoten mit eindeutigen MAC-IDs, die an Ph-Segmente angeschlossen sind, welche durch Ph-Repeater miteinander verbunden sind

3.4.34

Little-Endian

(en: little endian)

Modell der Speicherorganisation, bei dem das am wenigsten signifikante Oktett an der kleinsten Adresse gespeichert wird, und Modell der Übertragungsorganisation, bei dem das am wenigsten signifikante Oktett zuerst über das Medium übertragen wird

3.4.35

L-Paket

(Link-Paket)

klar abgegrenzter Unterabschnitt einer DLPDU, bestehend aus:

- a) Größenangaben und Steuerungsinformationen,
- b) einem festen Tag oder einem generischen Tag und
- c) DLS-Benutzerdaten oder, wenn das Tag DL-Signifikanz hat, DL-Daten

3.4.36

M_Zeichen

(en: M_symbol(s))

Symbol(e), das (die) die Datenbits und zugehörigen Informationen darstellt (darstellen), die von der Physical Layer codiert und übertragen werden

3.4.37**höchster geplanter Knoten**

(en: maximum scheduled node)

Knoten mit der höchsten MAC-ID, der geplante Zeit auf einem Verbindungszweig nutzen kann

3.4.38**höchster ungeplanter Knoten**

(en: maximum unscheduled node)

Knoten mit der höchsten MAC-ID, der ungeplante Zeit auf einem Verbindungszweig nutzen kann

3.4.39**Nachrichtenrouter**

(en: Message Router)

Objekt innerhalb eines Knotens, das Nachrichten Anforderungen an die entsprechenden Anwendungsobjekte verteilt

3.4.40**Moderator**

(en: moderator)

Knoten mit der niedrigsten MAC-ID, der für die Übertragung der Moderator-DLPDU verantwortlich ist

3.4.41**Moderator-DLPDU**

(en: moderator DLPDU)

DLPDU, die von dem Knoten mit der niedrigsten MAC-ID zum Zweck der Synchronisation der Knoten und der Verteilung der Konfigurationsparameter des Verbindungszweigs übertragen wird

3.4.42**Mehrpunktverbindung**

(en: multipoint connection)

Verbindung von einem Knoten zu vielen Knoten

Anmerkung 1 zum Begriff: Durch Mehrpunktverbindungen können Nachrichten von einem einzelnen Produzenten (Publisher) von vielen Konsumentenknoten (Subscribern) empfangen werden.

3.4.43**Netzwerk**

(en: network)

Reihe von Knoten, die über ein bestimmtes Kommunikationsmedium miteinander verbunden sind

Anmerkung 1 zum Begriff: Die Verbindungspfade zwischen beliebigen Paaren von Knoten können Repeater, Router und Gateways einschließen.

3.4.44**Network Access Monitor****NAM**

(en: network access monitor, NAM)

Komponente innerhalb eines Knotens, welche die Kommunikation mit einem temporären Knoten, der mit den lokalen NAPs der Knoten verbunden ist, steuert

3.4.45**Netzzugangspunkt****NAP**

(en: network access port, NAP)

Variante der Physical Layer, bei der ein temporärer Knoten mit dem Verbindungszweig durch Verbindung mit dem NAP eines permanenten Knotens verbunden werden kann

3.4.46

Netzwerkadresse oder Knotenadresse

(en: network address or node address)

Adresse eines Knotens auf dem Verbindungszweig

Anmerkung 1 zum Begriff: Wird auch MAC-ID genannt.

3.4.47

Netzwerkstatusanzeigen

(en: network status indicators)

Anzeigen an einem Knoten, die den Status des Physical Layer und des Data Link Layer anzeigen

3.4.48

NUT

(en: network update time, NUT)

sich wiederholendes Zeitintervall, in dem Daten auf dem Verbindungszweig gesendet werden können

3.4.49

Knoten

(en: node)

logische Verbindung zu einem lokalen Verbindungszweig, für die eine einzelne MAC-ID erforderlich ist

Anmerkung 1 zum Begriff: Ein einzelnes physisches Gerät kann in Form von vielen Knoten auf dem gleichen lokalen Verbindungszweig erscheinen. Für den Zweck dieses Protokolls wird jeder Knoten als einzelne DLE betrachtet.

3.4.50

Nichtübereinstimmung

(en: non-concurrence)

Situation, in der eine Übertragung von einer unerwarteten MAC-ID empfangen wird, die das zeitbasierte Zugangsprotokoll zu verletzen scheint

Anmerkung 1 zum Begriff: Eine Nichtübereinstimmung kann auftreten, wenn eine Verbindung zwischen zwei betriebsfähigen Verbindungszweigen, die nicht miteinander synchronisiert werden, aber über die gleichen Konfigurationsinformationen verfügen, hergestellt wird.

3.4.51

Nicht-Datensymbol

(en: non-data symbol)

Symbol der Physical Layer, das die Anforderungen der Manchester-Biphase-L-Codierung verletzt

3.4.52

Objekt

(en: object)

abstrakte Darstellung einer bestimmten Komponente innerhalb eines Gerätes

Anmerkung 1 zum Begriff: Ein Objekt kann sein:

- 1) eine abstrakte Darstellung der Fähigkeiten eines Computers. Objekte können aus einigen der folgenden oder aus allen folgenden Komponenten bestehen:
 - a) Daten (Informationen, die sich im Laufe der Zeit ändern);
 - b) Konfiguration (Parameter für das Verhalten);
 - c) Methoden (Dinge, die mithilfe der Daten und der Konfiguration durchgeführt werden können);
- 2) eine Zusammenstellung zugehöriger Daten (in Form von Variablen) und Methoden (Prozeduren) für Operationen mit diesen Daten, die eine klar definierte Schnittstelle und ein klar definiertes Verhalten aufweisen.

3.4.53

objektspezifischer Dienst

(en: object specific service)

durch eine bestimmte Objektklasse definierter Dienst, um eine erforderliche Funktion auszuführen, welche nicht durch einen allgemeinen Dienst ausgeführt wird

Anmerkung 1 zum Begriff: Ein objektspezifischer Dienst ist für die Objektklasse, die ihn definiert, eindeutig.

3.4.54

Absender

(en: originator)

Client, der für den Aufbau eines Verbindungspfads zum Ziel verantwortlich ist

3.4.55

permanenter Knoten

(en: permanent node)

Knoten, dessen Verbindung zum Netzwerk keinen Netzzugangspunkt (NAP) der Physical Layer-Variante verwendet

Anmerkung 1 zum Begriff: Dieser Knoten kann optional einen NAP der Physical Layer-Variante unterstützen, um die Verbindung temporärer Knoten mit dem Netzwerk zu ermöglichen.

3.4.56

produzieren

(en: produce)

Vorgang des Sendens von Daten zum Empfang durch einen Konsumenten

3.4.57

Produzent

(en: producer)

Knoten, der für das Senden von Daten verantwortlich ist

3.4.58

redundante Medien

(en: redundant media)

System, das mehr als ein Medium einsetzt, um Kommunikationsausfällen vorzubeugen

3.4.59

angefordertes Paketintervall

RPI

(en: requested packet interval, RPI)

Maß, wie häufig die absendende Anwendung die Übertragung von L-Paketen oder Datenpaketen von der Zielanwendung anfordert

3.4.60

Rogue

(en: rogue)

Knoten, der eine Moderator-DLPDU empfangen hat, die nicht mit der Konfiguration des Verbindungszweigs, die aktuell von diesem Knoten verwendet wird, übereinstimmt

3.4.61

geplant

(en: scheduled)

Datentransfers, die deterministisch und wiederholt zu vorgegebenen NUTs auftreten

3.4.62

Server

(en: server)

Objekt, das einem anderen (Client-)Objekt Dienste bereitstellt

3.4.63

Dienst

(en: service)

Operation oder Funktion, die ein Objekt und/oder eine Objektklasse auf Anforderung eines anderen Objekts und/oder einer anderen Objektklasse hin ausführt

Anmerkung 1 zum Begriff: Es sind eine Reihe von allgemeinen Diensten definiert und es sind Richtlinien für die Festlegung objektspezifischer Dienste vorgesehen. Objektspezifische Dienste sind Dienste, die durch eine bestimmte Objektklasse festgelegt sind, um eine erforderliche Funktion auszuführen, die nicht von einem allgemeinen Dienst ausgeführt wird.

3.4.64

Schlitzzeit

(en: slot time)

erforderliche maximale Dauer für die Erkennung einer erwarteten Übertragung

Anmerkung 1 zum Begriff: Während des Durchlaufs des implizierten Tokens wartet jeder Knoten für jeden fehlenden Knoten eine Schlitzzeit lang.

3.4.65

Startzeichen

(en: start delimiter)

eindeutiger Satz von M_Zeichen, der den Beginn einer DLPDU angibt

3.4.66

Superknoten

(en: supernode)

DLE mit einer MAC-ID von null

Anmerkung 1 zum Begriff: Die DL-Adresse dieser DLE ist für spezielle DLL-Funktionen reserviert.

3.4.67

eindeutige Tabellenkennung

TUI

(en: table unique identifier, TUI)

eindeutiger Referenzcode, der von ControlNet- und Keeper-Objekten für die Referenzierung einer konsistenten Menge von Konfigurationsattributen des Verbindungszweigs verwendet wird

3.4.68

Tag

(en: tag)

Abkürzung für einen bestimmten Teil von Anwendungsinformationen, zwei oder drei Oktette lang

3.4.69

Ziel

(en: target)

Endknoten, mit dem eine Verbindung aufgebaut wird

3.4.70

temporärer Knoten

(en: temporary node)

entspricht einem transientem Knoten

3.4.71

tMinus

(en: tMinus)

Anzahl von NUTs, bevor ein neuer Satz von Konfigurationsparametern verwendet werden muss

3.4.72

Tone

(en: tone)

Zeitpunkt, der die Grenze zwischen zwei NUTs markiert

3.4.73

Tool

(en: tool)

ausführbares Softwareprogramm, das mit dem Benutzer interagiert, um bestimmte Funktionen auszuführen

BEISPIEL Software zur Planung des Verbindungszweigs.

3.4.74

Transaktions-ID

(en: transaction id)

Feld innerhalb eines UCMM-Headers, durch das eine Antwort der zugehörigen Anforderung zugeordnet wird und das ein Server in der Antwortnachricht wiederholt

3.4.75

transienter Knoten

(en: transient node)

Knoten, der nur temporär mithilfe des NAP des Physical Layer-Mediums, das mit dem NAP eines permanenten Knotens verbunden ist, mit dem Netzwerk verbunden werden soll

3.4.76

Unconnected Message Manager

UCMM

(en: Unconnected Message Manager, UCMM)

Komponente innerhalb eines Knotens, die verbindungslose, explizite Nachrichten überträgt und empfängt und direkt an das Nachrichtenrouter-Objekt sendet

3.4.77

verbindungsloser Dienst

(en: unconnected service)

Nachrichtendienst, der nicht auf den Aufbau einer Verbindung zwischen Geräten angewiesen ist, bevor Informationen ausgetauscht werden können

3.4.78

ungeplant

(en: unscheduled)

Datentransfers, die die verbleibende Zeit in der NUT nach Abschluss der planmäßigen Transfers nutzen

3.4.79

Anbieter-ID

(en: vendor ID)

Kennzeichnung jedes Produktherstellers/Anbieters durch eine eindeutige Nummer

Anmerkung 1 zum Begriff: Anbieter-IDs werden von der ODVA, Inc. vergeben.

– Leerseite –

English Version

Industrial communication networks - Fieldbus specifications –
Part 4-2: Data-link layer protocol specification - Type 2 elements
(IEC 61158-4-2:2014)

Réseaux de communication industriels - Spécifications des
bus de terrain - Partie 4-2: Spécification du protocole de la
couche liaison de données - Éléments de type 2
(CEI 61158-4-2:2014)

Industrielle Kommunikationsnetze - Feldbusse - Teil 4-2:
Protokollspezifikation des Data Link Layer
(Sicherheitsschicht) - Typ 2-Elemente
(IEC 61158-4-2:2014)

This European Standard was approved by CENELEC on 2014-09-19. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, Former Yugoslav Republic of Macedonia, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.



European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Avenue Marnix 17, B-1000 Brussels

Foreword

The text of document 65C/762/FDIS, future edition 3 of IEC 61158-4-2, prepared by SC 65C "Industrial networks" of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN 61158-4-2:2014.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2015-06-19
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2017-09-19

This document supersedes EN 61158-4-2:2012.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC [and/or CEN] shall not be held responsible for identifying any or all such patent rights.

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

Endorsement notice

The text of the International Standard IEC 61158-4-2:2014 was approved by CENELEC as a European Standard without any modification.

In the official version, for bibliography, the following notes have to be added for the standards indicated:

IEC 61158-1:2014	NOTE	Harmonised as EN 61158-1:2014
IEC 61158-2:2014	NOTE	Harmonised as EN 61158-2:2014
IEC 61784-1:2014	NOTE	Harmonised as EN 61784-1:2014
IEC 61784-2:2014	NOTE	Harmonised as EN 61784-2:2014

CONTENTS

INTRODUCTION.....	11
1 Scope	13
1.1 General.....	13
1.2 Specifications	13
1.3 Procedures.....	13
1.4 Applicability	14
1.5 Conformance.....	14
2 Normative references	14
3 Terms, definitions, symbols, abbreviations and conventions	15
3.1 Reference model terms and definitions.....	15
3.2 Service convention terms and definitions.....	17
3.3 Common terms and definitions	18
3.4 Additional Type 2 definitions	20
3.5 Type 2 symbols and abbreviations	27
3.6 Conventions for station management objects.....	28
4 Overview of the data-link protocol	29
4.1 General.....	29
4.2 Services provided by the DL	31
4.3 Structure and definition of DL-addresses	32
4.4 Services assumed from the PhL	34
4.5 Functional classes	37
5 General structure and encoding of PhIDUs and DLPDUs and related elements of procedure.....	37
5.1 Overview	37
5.2 Media access procedure	37
5.3 DLPDU structure and encoding	41
5.4 Lpacket components.....	45
5.5 DLPDU procedures.....	47
5.6 Summary of DLL support services and objects	48
6 Specific DLPDU structure, encoding and procedures	50
6.1 Modeling language	50
6.2 DLS user services	52
6.3 Generic tag Lpacket.....	58
6.4 Moderator Lpacket.....	58
6.5 Time distribution Lpacket.....	60
6.6 UCMM Lpacket.....	62
6.7 Keeper UCMM Lpacket	62
6.8 TUI Lpacket.....	63
6.9 Link parameters Lpacket and tMinus Lpacket	64
6.10 I'm-alive Lpacket	66
6.11 Ping Lpackets.....	67
6.12 WAMI Lpacket	68
6.13 Debug Lpacket	69
6.14 IP Lpacket.....	70
6.15 Ethernet Lpacket	70
7 Objects for station management.....	70

7.1	General	70
7.2	ControlNet object	71
7.3	Keeper object	82
7.4	Scheduling object	103
7.5	TCP/IP Interface object	114
7.6	Ethernet link object	134
7.7	DeviceNet object	143
7.8	Connection configuration object (CCO)	151
7.9	DLR object	173
7.10	QoS object	188
7.11	Port object	190
8	Other DLE elements of procedure	194
8.1	Network attachment monitor (NAM)	194
8.2	Calculating link parameters	201
9	Detailed specification of DL components	208
9.1	General	208
9.2	Access control machine (ACM)	208
9.3	TxLLC	225
9.4	RxLLC	229
9.5	Transmit machine (TxM)	232
9.6	Receive machine (RxM)	236
9.7	Serializer	242
9.8	Deserializer	244
9.9	DLL management	244
10	Device Level Ring (DLR) protocol	246
10.1	General	246
10.2	Supported topologies	247
10.3	Overview of DLR operation	248
10.4	Classes of DLR implementation	251
10.5	DLR behavior	251
10.6	Implementation requirements	257
10.7	Using non-DLR nodes in the ring network	259
10.8	Redundant gateway devices on DLR network	262
10.9	DLR messages	267
10.10	State diagrams and state-event-action matrices	273
10.11	Performance analysis	300
Annex A (normative)	Indicators and switches	306
A.1	Purpose	306
A.2	Indicators	306
A.2.1	General indicator requirements	306
A.2.2	Common indicator requirements	306
A.2.3	Fieldbus specific indicator requirements (1)	308
A.2.4	Fieldbus specific indicator requirements (2)	312
A.2.5	Fieldbus specific indicator requirements (3)	315
A.3	Switches	319
A.3.1	Common switch requirements	319
A.3.2	Fieldbus specific switch requirements (1)	320

A.3.3	Fieldbus specific switch requirements (2)	320
A.3.4	Fieldbus specific switch requirements (3)	321
	Bibliography	322
Annex ZA (normative)	Normative references to international publications with their corresponding European publications	324
Figure 1	– Relationships of DLSAPs, DLSAP-addresses and group DL-addresses	19
Figure 2	– Data-link layer internal architecture	30
Figure 3	– Basic structure of a MAC ID address	32
Figure 4	– Basic structure of a generic tag address	33
Figure 5	– Basic structure of a fixed tag address	33
Figure 6	– M_symbols and Manchester encoding at 5 MHz	35
Figure 7	– NUT structure	38
Figure 8	– Media access during scheduled time	39
Figure 9	– Media access during unscheduled time	40
Figure 10	– DLPDU format	41
Figure 11	– Aborting a DLPDU during transmission	45
Figure 12	– Lpacket format	45
Figure 13	– Generic tag Lpacket format	46
Figure 14	– Fixed tag Lpacket format	47
Figure 15	– Goodness parameter of TimeDist_Lpacket	60
Figure 16	– Example I'm alive processing algorithm	67
Figure 17	– Keeper CRC algorithm	88
Figure 18	– Keeper object power-up state diagram	99
Figure 19	– Keeper object operating state diagram	100
Figure 20	– Synchronized network change processing	103
Figure 21	– State transition diagram for TCP/IP Interface object	127
Figure 22	– State transition diagram for TCP/IP Interface object (continued)	128
Figure 23	– ACD Behavior	130
Figure 24	– State transition diagram for Ethernet Link object	143
Figure 25	– Connection configuration object edit flowchart	173
Figure 26	– NAM state machine	195
Figure 27	– DLR rings connected to switches	247
Figure 28	– Normal operation of a DLR network	248
Figure 29	– Beacon and Announce frames	249
Figure 30	– Link failure	250
Figure 31	– Network reconfiguration after link failure	250
Figure 32	– Neighbor Check process	257
Figure 33	– Unsupported topology – example 1	260
Figure 34	– Unsupported topology – example 2	261
Figure 35	– DLR ring connected to switches through redundant gateways	263
Figure 36	– DLR redundant gateway capable device	264
Figure 37	– Advertise frame	266

Figure 38 – State transition diagram for Beacon frame based non-supervisor ring node	273
Figure 39 – State transition diagram for Announce frame based non-supervisor ring node..	279
Figure 40 – State transition diagram for ring supervisor	283
Figure 41 – State transition diagram for redundant gateway.....	296
Figure A.1 – Non redundant network status indicator labeling	312
Figure A.2 – Redundant network status indicator labeling	312
Figure A.3 – Network status indicator state diagram	315

Table 1 – Format of attribute tables.....	28
Table 2 – Data-link layer components	30
Table 3 – MAC ID addresses allocation.....	33
Table 4 – Fixed tag service definitions	34
Table 5 – Data encoding rules	35
Table 6 – M Data symbols	36
Table 7 – Truth table for ph_status_indication	36
Table 8 – FCS length, polynomials and constants.....	42
Table 9 – DLL support services and objects	49
Table 10 – Elementary data types	52
Table 11 – DLL events.....	56
Table 12 – Time distribution priority	61
Table 13 – Format of the TUI Lpacket	64
Table 14 – ControlNet object class attributes	72
Table 15 – ControlNet object instance attributes.....	72
Table 16 – TUI status flag bits	76
Table 17 – Mac_ver bits.....	77
Table 18 – Channel state bits	77
Table 19 – ControlNet object common services	79
Table 20 – ControlNet object class specific services.....	80
Table 21 – Keeper object revision history	82
Table 22 – Keeper object class attributes	82
Table 23 – Keeper object instance attributes	83
Table 24 – Keeper operating state definitions.....	86
Table 25 – Port status flag bit definitions.....	86
Table 26 – TUI status flag bits	87
Table 27 – Keeper attributes.....	89
Table 28 – Memory requirements (in octets) for the Keeper attributes.....	90
Table 29 – Keeper object common services.....	90
Table 30 – Keeper object class specific services	91
Table 31 – Service error codes	92
Table 32 – Wire order format of the TUI Lpacket	96
Table 33 – Service error codes	97
Table 34 – Keeper object operating states	97

Table 35 – Keeper object state event matrix	101
Table 36 – Scheduling object class attributes	104
Table 37 – Scheduling object instance attributes	105
Table 38 – Scheduling object common services	105
Table 39 – Status error descriptions for Create	106
Table 40 – Status error descriptions for Delete and Kick_Timer	107
Table 41 – Scheduling object class specific services	107
Table 42 – Status error descriptions for Read	109
Table 43 – Status error descriptions for Conditional_Write	110
Table 44 – Status error descriptions for Forced_Write	110
Table 45 – Status error descriptions for Change_Start	111
Table 46 – Status error descriptions for Break_Connections	111
Table 47 – Status error descriptions for Change_Complete	112
Table 48 – Status error descriptions for Restart_Connections	112
Table 49 – Revision history	114
Table 50 – TCP/IP Interface object class attributes	114
Table 51 – TCP/IP Interface object instance attributes	115
Table 52 – Status bits	118
Table 53 – Configuration capability bits	119
Table 54 – Configuration control bits	119
Table 55 – Example path	120
Table 56 – Interface configuration components	121
Table 57 – Alloc control values	123
Table 58 – AcdActivity values	124
Table 59 – ArpPdu - ARP Response PDU in binary format	124
Table 60 – TCP/IP Interface object common services	125
Table 61 – Get_Attribute_All reply format	126
Table 62 – Ethernet link object revision history	134
Table 63 – Ethernet link object class attributes	135
Table 64 – Ethernet link object instance attributes	135
Table 65 – Interface flags bits	138
Table 66 – Control bits	139
Table 67 – Interface type	140
Table 68 – Interface state	140
Table 69 – Admin state	140
Table 70 – Ethernet Link object common services	141
Table 71 – Get_Attribute_All reply format	142
Table 72 – Ethernet Link object class specific services	142
Table 73 – DeviceNet object revision history	144
Table 74 – DeviceNet object class attributes	144
Table 75 – DeviceNet object instance attributes	144
Table 76 – Bit rate attribute values	146

Table 77 – BOI attribute values	147
Table 78 – Diagnostic counters bit description.....	149
Table 79 – DeviceNet object common services.....	150
Table 80 – Reset service parameter.....	150
Table 81 – Reset service parameter values.....	150
Table 82 – DeviceNet object class specific services	151
Table 83 – Connection configuration object revision history	152
Table 84 – Connection configuration object class attributes	152
Table 85 – Format number values	153
Table 86 – Connection configuration object instance attributes	154
Table 87 – Originator connection status values	158
Table 88 – Target connection status values.....	158
Table 89 – Connection flags	158
Table 90 – I/O mapping formats	160
Table 91 – Services valid during a change operation	162
Table 92 – Connection configuration object common services.....	163
Table 93 – Get_Attribute_All Response – class level	163
Table 94 – Get_Attribute_All response – instance level	164
Table 95 – Set_Attribute_All error codes	165
Table 96 – Set_Attribute_All request.....	166
Table 97 – Create request parameters	168
Table 98 – Create error codes	168
Table 99 – Delete error codes.....	168
Table 100 – Restore error codes.....	169
Table 101 – Connection configuration object class specific services	169
Table 102 – Change_Start error codes.....	170
Table 103 – Get_Status service parameter.....	171
Table 104 – Get_Status service response	171
Table 105 – Get_Status service error codes.....	171
Table 106 – Change_Complete service parameter.....	172
Table 107 – Change_Complete service error codes.....	172
Table 108 – Audit_Changes service parameter.....	172
Table 109 – Audit_Changes service error codes.....	172
Table 110 – Revision history	174
Table 111 – DLR object class attributes	174
Table 112 – DLR object instance attributes	174
Table 113 – Network Status values	178
Table 114 – Ring Supervisor Status values	178
Table 115 – Capability flags.....	181
Table 116 – Redundant Gateway Status values.....	183
Table 117 – DLR object common services.....	184
Table 118 – Get_Attribute_All Response – Object Revision 1, non supervisor device	184

Table 119 – Get_Attribute_All Response – Object Revision 1, supervisor-capable device...	185
Table 120 – Get_Attribute_All Response – Object Revision 2, non supervisor device	185
Table 121 – Get_Attribute_All Response – All other cases.....	186
Table 122 – DLR object class specific services.....	187
Table 123 – QoS object revision history	188
Table 124 – QoS object class attributes	188
Table 125 – QoS object instance attributes	189
Table 126 – Default DCSP values and usages.....	190
Table 127 – QoS object common services	190
Table 128 – Port object class attributes.....	191
Table 129 – Port object instance attributes.....	191
Table 130 – Port object common services	193
Table 131 – NAM states.....	194
Table 132 – Default link parameters	195
Table 133 – PhL timing characteristics	202
Table 134 – DLR variables.....	252
Table 135 – Redundant gateway variables	265
Table 136 – MAC addresses for DLR messages	267
Table 137 – IEEE 802.1Q common frame header format.....	268
Table 138 –DLR message payload fields.....	268
Table 139 – DLR frame types.....	268
Table 140 – Format of the Beacon frame.....	269
Table 141 – Ring State values	269
Table 142 – Format of the Neighbor_Check request	269
Table 143 – Format of the Neighbor_Check response.....	270
Table 144 – Format of the Link_Status/Neighbor_Status frame	270
Table 145 – Link/Neighbor status values	270
Table 146 – Format of the Locate_Fault frame.....	271
Table 147 – Format of the Announce frame	271
Table 148 – Format of the Sign_On frame	271
Table 149 – Format of the Advertise frame	272
Table 150 – Gateway state values.....	272
Table 151 – Format of the Flush_Tables frame.....	272
Table 152 – Format of the Learning_Update frame	273
Table 153 – Parameter values for Beacon frame based non-supervisor ring node	274
Table 154 – LastBcnRcvPort bit definitions.....	274
Table 155 – State-event-action matrix for Beacon frame based non-supervisor ring node ...	274
Table 156 – Parameter values for Announce frame based non-supervisor ring node.....	279
Table 157 – State-event-action matrix for Announce frame based non-supervisor ring node	280
Table 158 – Parameter values for ring supervisor node.....	284
Table 159 – LastBcnRcvPort bit definitions.....	285
Table 160 – State-event-action matrix for ring supervisor node	285

Table 161 – Parameter values for redundant gateway node	297
Table 162 – State-event-action matrix for redundant gateway node.....	298
Table 163 – Parameters/assumptions for example performance calculations.....	300
Table 164 – Example ring configuration parameters and performance.....	303
Table 165 – Variables for performance analysis	304
Table A.1 – Module status indicator	307
Table A.2 – Time Sync status indication.....	308
Table A.3 – Network status indicators	310
Table A.4 – Network status indicator	314
Table A.5 – Network status indicator	317
Table A.6 – Combined module/network status indicator	318
Table A.7 – I/O status indicator	319
Table A.8 – Bit rate switch encoding	321

INTRODUCTION

This part of IEC 61158 is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the “three-layer” fieldbus reference model described in IEC 61158-1.

The data-link protocol provides the data-link service by making use of the services available from the physical layer. The primary aim of this standard is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer data-link entities (DLEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

- a) as a guide for implementers and designers;
- b) for use in the testing and procurement of equipment;
- c) as part of an agreement for the admittance of systems into the open systems environment;
- d) as a refinement to the understanding of time-critical communications within OSI.

This standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this standard together with other standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems may work together in any combination.

The International Electrotechnical Commission (IEC) draws attention to the fact that it is claimed that compliance with this document may involve the use of patents given in several subclauses as indicated in the table below. These patents are held by their respective inventors under license to ODVA, Inc:

US 5,400,331	[ODVA]	Communication network interface with screeners for incoming messages	Subclause 3.4, Clauses 4 to 9
US 5,471,461	[ODVA]	Digital communication network with a moderator station election process	
US 5,491,531	[ODVA]	Media access controller with a shared class message delivery capability	
US 5,493,571	[ODVA]	Apparatus and method for digital communications with improved delimiter detection	
US 5,537,549	[ODVA]	Communication network with time coordinated station activity by time slot and periodic interval number	
US 5,553,095	[ODVA]	Method and apparatus for exchanging different classes of data during different time intervals	Clause 10
US 8,244,838	[ODVA]	Industrial controller employing the network ring topology	

IEC takes no position concerning the evidence, validity and scope of these patent rights.

ODVA and the holders of these patent rights have assured the IEC that ODVA is willing to negotiate licences either free of charge or under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of ODVA and the holders of these patent rights is registered with IEC. Information may be obtained from:

[ODVA] ODVA, Inc.
2370 East Stadium Boulevard #1000
Ann Arbor, Michigan 48104
USA
Attention: Office of the Executive Director
e-mail: odva@odva.org

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those identified above. IEC shall not be held responsible for identifying any or all such patent rights.

ISO (www.iso.org/patents) and IEC (<http://patents.iec.ch>) maintain on-line databases of patents relevant to their standards. Users are encouraged to consult the databases for the most up to date information concerning patents.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 4-2: Data-link layer protocol specification – Type 2 elements

1 Scope

1.1 General

The data-link layer provides basic time-critical messaging communications between devices in an automation environment.

This protocol provides communication opportunities to all participating data-link entities, sequentially and in a cyclic synchronous manner. Foreground scheduled access is available for time-critical activities together with background unscheduled access for less critical activities.

Deterministic and synchronized transfers can be provided at cyclic intervals up to 1 ms and device separations of 25 km. This performance is adjustable dynamically and on-line by re-configuring the parameters of the local link whilst normal operation continues. By similar means, DL connections and new devices may be added or removed during normal operation.

This protocol provides means to maintain clock synchronization across an extended link with a precision better than 10 μ s.

This protocol optimizes each access opportunity by concatenating multiple DLSDUs and associated DLPCI into a single DLPDU, thereby improving data transfer efficiency for data-link entities that actively source multiple streams of data.

The maximum system size is an unlimited number of links of 99 nodes, each with 255 DLSAP-addresses. Each link has a maximum of 2^{24} related peer and publisher DLCEPs.

1.2 Specifications

This standard specifies

- a) procedures for the timely transfer of data and control information from one data-link user entity to a peer user entity, and among the data-link entities forming the distributed data-link service provider;
- b) the structure of the fieldbus DLPDUs used for the transfer of data and control information by the protocol of this standard, and their representation as physical interface data units.

1.3 Procedures

The procedures are defined in terms of

- a) the interactions between peer DL-entities (DLEs) through the exchange of fieldbus DLPDUs;
- b) the interactions between a DL-service (DLS) provider and a DLS-user in the same system through the exchange of DLS primitives;
- c) the interactions between a DLS-provider and a Ph-service provider in the same system through the exchange of Ph-service primitives.

1.4 Applicability

These procedures are applicable to instances of communication between systems which support time-critical communications services within the data-link layer of the OSI or fieldbus reference models, and which require the ability to interconnect in an open systems interconnection environment.

Profiles provide a simple multi-attribute means of summarizing an implementation's capabilities, and thus its applicability to various time-critical communications needs.

1.5 Conformance

This standard also specifies conformance requirements for systems implementing these procedures. This standard does not contain tests to demonstrate compliance with such requirements.

2 Normative references

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

NOTE All parts of the IEC 61158 series, as well as IEC 61784-1 and IEC 61784-2 are maintained simultaneously. Cross-references to these documents within the text therefore refer to the editions as dated in this list of normative references.

IEC 61131-3, *Programmable controllers – Part 3: Programming languages*

IEC 61158-3-2:2014, *Industrial communication networks – Fieldbus specifications – Part 3-2: Data-link layer service definition – Type 2 elements*

IEC 61158-5-2:2014, *Industrial communication networks – Fieldbus specifications – Part 5-2: Application layer service definition – Type 2 elements*

IEC 61158-6-2:2014, *Industrial communication networks – Fieldbus specifications – Part 6-2: Application layer protocol specification – Type 2 elements*

IEC 61588:2009, *Precision clock synchronization protocol for networked measurement and control systems*

IEC 61784-3-2, *Industrial communication networks – Profiles – Part 3-2: Functional safety fieldbuses – Additional specifications for CPF 2*

IEC 62026-3:2008, *Low-voltage switchgear and controlgear – Controller-device interfaces (CDIs) – Part 3: DeviceNet*

ISO/IEC 3309¹, *Information technology – Telecommunications and information exchange between systems – High-level data link control (HDLC) procedures – Frame structure*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 7498-3, *Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing*

¹ This standard has been withdrawn.

ISO/IEC 8802-3, *Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications*

ISO 11898:1993², *Road vehicles – Interchange of digital information – Controller area network (CAN) for high-speed communication*

IEEE 802.1D-2004, *IEEE standard for local and metropolitan area networks – Media Access Control (MAC) bridges*, available at <<http://www.ieee.org>>

IEEE 802.1Q-2005², *IEEE standard for local and metropolitan area networks – Virtual bridged local area networks*, available at <<http://www.ieee.org>>

IEEE 802.3-2008, *IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications*, available at <<http://www.ieee.org>>

IETF RFC 951, *Bootstrap Protocol (BOOTP)*, available at <<http://www.ietf.org>>

IETF RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets: MIB-II*, available at <<http://www.ietf.org>>

IETF RFC 1542, *Clarifications and Extensions for the Bootstrap Protocol*, available at <<http://www.ietf.org>>

IETF RFC 1643, *Definitions of Managed Objects for the Ethernet-like Interface Types*, available at <<http://www.ietf.org>>

IETF RFC 2131, *Dynamic Host Configuration Protocol*, available at <<http://www.ietf.org>>

IETF RFC 2132, *DHCP Options and BOOTP Vendor Extensions*, available at <<http://www.ietf.org>>

IETF RFC 4541, *Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches*, available at <<http://www.ietf.org>>

IETF RFC 5227:2008, *IPv4 Address Conflict Detection*, available at <<http://www.ietf.org>>

3 Terms, definitions, symbols, abbreviations and conventions

For the purposes of this document, the following terms, definitions, symbols and abbreviations apply.

3.1 Reference model terms and definitions

This standard is based in part on the concepts developed in ISO/IEC 7498-1 and ISO/IEC 7498-3, and makes use of the following terms defined therein.

3.1.1 called-DL-address [ISO/IEC 7498-3]

3.1.2 calling-DL-address [ISO/IEC 7498-3]

² A newer edition of this standard has been published, but only the cited edition applies.

3.1.3	centralized multi-end-point-connection	[ISO/IEC 7498-1]
3.1.4	correspondent (N)-entities correspondent DL-entities (N=2) correspondent Ph-entities (N=1)	[ISO/IEC 7498-1]
3.1.5	demultiplexing	[ISO/IEC 7498-1]
3.1.6	DL-address	[ISO/IEC 7498-3]
3.1.7	DL-address-mapping	[ISO/IEC 7498-1]
3.1.8	DL-connection	[ISO/IEC 7498-1]
3.1.9	DL-connection-end-point	[ISO/IEC 7498-1]
3.1.10	DL-connection-end-point-identifier	[ISO/IEC 7498-1]
3.1.11	DL-connection-mode transmission	[ISO/IEC 7498-1]
3.1.12	DL-connectionless-mode transmission	[ISO/IEC 7498-1]
3.1.13	DL-data-sink	[ISO/IEC 7498-1]
3.1.14	DL-data-source	[ISO/IEC 7498-1]
3.1.15	DL-duplex-transmission	[ISO/IEC 7498-1]
3.1.16	DL-facility	[ISO/IEC 7498-1]
3.1.17	DL-local-view	[ISO/IEC 7498-3]
3.1.18	DL-name	[ISO/IEC 7498-3]
3.1.19	DL-protocol	[ISO/IEC 7498-1]
3.1.20	DL-protocol-connection-identifier	[ISO/IEC 7498-1]
3.1.21	DL-protocol-control-information	[ISO/IEC 7498-1]
3.1.22	DL-protocol-data-unit	[ISO/IEC 7498-1]
3.1.23	DL-protocol-version-identifier	[ISO/IEC 7498-1]
3.1.24	DL-relay	[ISO/IEC 7498-1]
3.1.25	DL-service-connection-identifier	[ISO/IEC 7498-1]
3.1.26	DL-service-data-unit	[ISO/IEC 7498-1]
3.1.27	DL-simplex-transmission	[ISO/IEC 7498-1]
3.1.28	DL-subsystem	[ISO/IEC 7498-1]
3.1.29	DL-user-data	[ISO/IEC 7498-1]
3.1.30	flow control	[ISO/IEC 7498-1]
3.1.31	layer-management	[ISO/IEC 7498-1]
3.1.32	multiplexing	[ISO/IEC 7498-3]
3.1.33	naming-(addressing)-authority	[ISO/IEC 7498-3]

3.1.34	naming-(addressing)-domain	[ISO/IEC 7498-3]
3.1.35	naming-(addressing)-subdomain	[ISO/IEC 7498-3]
3.1.36	(N)-entity DL-entity Ph-entity	[ISO/IEC 7498-1]
3.1.37	(N)-interface-data-unit DL-service-data-unit (N=2) Ph-interface-data-unit (N=1)	[ISO/IEC 7498-1]
3.1.38	(N)-layer DL-layer (N=2) Ph-layer (N=1)	[ISO/IEC 7498-1]
3.1.39	(N)-service DL-service (N=2) Ph-service (N=1)	[ISO/IEC 7498-1]
3.1.40	(N)-service-access-point DL-service-access-point (N=2) Ph-service-access-point (N=1)	[ISO/IEC 7498-1]
3.1.41	(N)-service-access-point-address DL-service-access-point-address (N=2) Ph-service-access-point-address (N=1)	[ISO/IEC 7498-1]
3.1.42	peer-entities	[ISO/IEC 7498-1]
3.1.43	Ph-interface-control-information	[ISO/IEC 7498-1]
3.1.44	Ph-interface-data	[ISO/IEC 7498-1]
3.1.45	primitive name	[ISO/IEC 7498-3]
3.1.46	reassembling	[ISO/IEC 7498-1]
3.1.47	recombining	[ISO/IEC 7498-1]
3.1.48	reset	[ISO/IEC 7498-1]
3.1.49	responding-DL-address	[ISO/IEC 7498-3]
3.1.50	routing	[ISO/IEC 7498-1]
3.1.51	segmenting	[ISO/IEC 7498-1]
3.1.52	sequencing	[ISO/IEC 7498-1]
3.1.53	splitting	[ISO/IEC 7498-1]
3.1.54	synonymous name	[ISO/IEC 7498-3]
3.1.55	systems-management	[ISO/IEC 7498-1]

3.2 Service convention terms and definitions

This standard also makes use of the following terms defined in ISO/IEC 10731 as they apply to the data-link layer:

- 3.2.1 acceptor
- 3.2.2 asymmetrical service
- 3.2.3 confirm (primitive);
requestor.deliver (primitive)
- 3.2.4 deliver (primitive)
- 3.2.5 DL-confirmed-facility
- 3.2.6 DL-facility
- 3.2.7 DL-local-view
- 3.2.8 DL-mandatory-facility
- 3.2.9 DL-non-confirmed-facility
- 3.2.10 DL-provider-initiated-facility
- 3.2.11 DL-provider-optional-facility
- 3.2.12 DL-service-primitive;
primitive
- 3.2.13 DL-service-provider
- 3.2.14 DL-service-user
- 3.2.15 DL-user-optional-facility
- 3.2.16 indication (primitive)
acceptor.deliver (primitive)
- 3.2.17 multi-peer
- 3.2.18 request (primitive);
requestor.submit (primitive)
- 3.2.19 requestor
- 3.2.20 response (primitive);
acceptor.submit (primitive)
- 3.2.21 submit (primitive)
- 3.2.22 symmetrical service

3.3 Common terms and definitions

For the purposes of this document, the following terms and definitions apply.

NOTE Many definitions are common to more than one protocol Type; they are not necessarily used by all protocol Types.

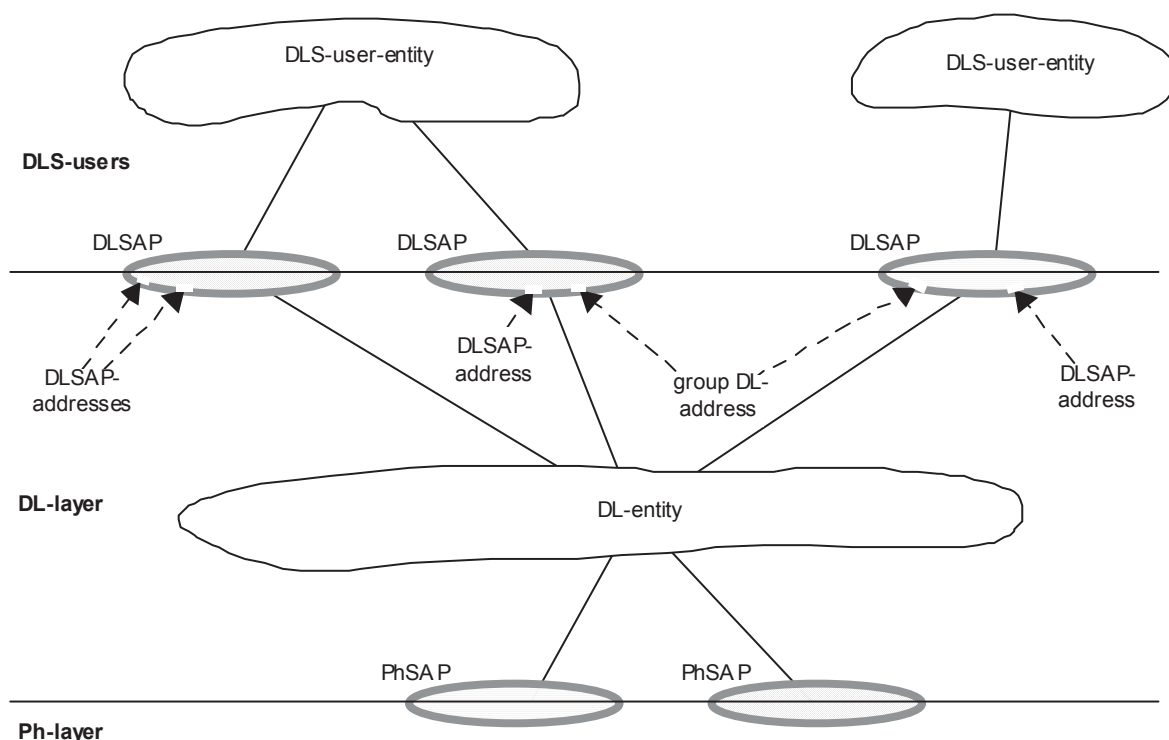
3.3.1

DL-segment

link

local link

single DL-subnetwork in which any of the connected DLEs may communicate directly, without any intervening DL-relaying, whenever all of those DLEs that are participating in an instance of communication are simultaneously attentive to the DL-subnetwork during the period(s) of attempted communication



NOTE 1 DLSAPs and PhSAPs are depicted as ovals spanning the boundary between two adjacent layers.

NOTE 2 DL-addresses are depicted as designating small gaps (points of access) in the DLL portion of a DLSAP.

NOTE 3 A single DL-entity can have multiple DLSAP-addresses and group DL-addresses associated with a single DLSAP.

Figure 1 – Relationships of DLSAPs, DLSAP-addresses and group DL-addresses

3.3.2

DLSAP

distinctive point at which DL-services are provided by a single DL-entity to a single higher-layer entity

Note 1 to entry: This definition, derived from ISO/IEC 7498-1, is repeated here to facilitate understanding of the critical distinction between DLSAPs and their DL-addresses. (See Figure 1.).

3.3.3

DL(SAP)-address

either an individual DLSAP-address, designating a single DLSAP of a single DLS-user, or a group DL-address potentially designating multiple DLSAPs, each of a single DLS-user

Note 1 to entry: This terminology is chosen because ISO/IEC 7498-3 does not permit the use of the term DLSAP-address to designate more than a single DLSAP at a single DLS-user.

3.3.4

(individual) DLSAP-address

DL-address that designates only one DLSAP within the extended link

Note 1 to entry: A single DL-entity may have multiple DLSAP-addresses associated with a single DLSAP.

3.3.5

extended link

DL-subnetwork, consisting of the maximal set of links interconnected by DL-relays, sharing a single DL-name (DL-address) space, in which any of the connected DL-entities may communicate, one with another, either directly or with the assistance of one or more of those intervening DL-relay entities

Note 1 to entry: An extended link may be composed of just a single link.

3.3.6

frame

denigrated synonym for DLPDU

3.3.7

group DL-address

DL-address that potentially designates more than one DLSAP within the extended link. A single DL-entity may have multiple group DL-addresses associated with a single DLSAP. A single DL-entity also may have a single group DL-address associated with more than one DLSAP

3.3.8

node

single DL-entity as it appears on one local link

3.3.9

receiving DLS-user

DL-service user that acts as a recipient of DL-user-data

Note 1 to entry: A DL-service user can be concurrently both a sending and receiving DLS-user.

3.3.10

sending DLS-user

DL-service user that acts as a source of DL-user-data

3.4 Additional Type 2 definitions

3.4.1

actual packet interval

API

measure of how frequently a specific connection produces its Lpacket data

3.4.2

allocate

take a resource from a common area and assign that resource for the exclusive use of a specific entity

3.4.3

application

function or data structure for which data is consumed or produced

3.4.4

application objects

multiple object classes that manage and provide the run time exchange of messages across the network and within the device

3.4.5

attribute

description of an externally visible characteristic or feature of an object

Note 1 to entry: The attributes of an object contain information about variable portions of an object. Typically, they provide status information or govern the operation of an object. Attributes may also affect the behavior of an object. Attributes are divided into class attributes and instance attributes.

3.4.6

behavior

indication of how the object responds to particular events

Note 1 to entry: Its description includes the relationship between attribute values and services.

3.4.7

bit

unit of information consisting of a 1 or a 0

Note 1 to entry: This is the smallest data unit that can be transmitted.

3.4.8

blanking or blanking time

length of time required after transmitting before the node is allowed to receive

3.4.9

client

- 1) An object which uses the services of another (server) object to perform a task
- 2) An initiator of a message to which a server reacts

3.4.10

communication objects

components that manage and provide run time exchange of messages across the network such as the ControlNet object, the Unconnected Message Manager (UCMM)

3.4.11

connection

logical binding between two application objects within the same or different devices

3.4.12

connection ID

CID

identifier assigned to a transmission, associated with a particular connection between producers and consumers, that identifies a specific piece of application information

Note 1 to entry: Within the data link this is a generic tag.

3.4.13

cyclic redundancy check

CRC

residual value computed from an array of data and used as a representative signature for the array

3.4.14

cyclic

term used to describe events which repeat in a regular and repetitive manner

3.4.15

deafness

situation where the node can not hear the moderator DLPDU but can hear other link traffic

3.4.16

device

physical hardware connection to the link

Note 1 to entry: A device may contain more than one node.

3.4.17

DLPDU

data-link protocol data unit

Note 1 to entry: A DLPDU consists of a source MAC ID, zero or more Lpackets and an FCS as transmitted or received by an associated PhE.

3.4.18

end delimiter

unique set of M_symbols that identifies the end of a DLPDU

3.4.19

error

discrepancy between a computed, observed or measured value or condition and the specified or theoretically correct value or condition

3.4.20

extended link

links connected by DL-routers (sometimes called a local area network)

3.4.21

FCS error

error that occurs when the computed frame check sequence value after reception of all the octets in a DLPDU does not match the expected residual

3.4.22

fixed tag

two octet identifier (tag) which identifies a specific service to be performed by either

- a) that receiving node on the local link which has a specified MAC ID, or
- b) all receiving nodes on the local link

Note 1 to entry: Identification of the target node(s) is included in the two octet tag.

3.4.23

frame

denigrated synonym for DLPDU

3.4.24

Frame Check Sequence

FCS

redundant data derived from a block of data within a DLPDU (frame), using a hash function, and stored or transmitted together with the block of data, in order to detect data corruption

3.4.25

generic tag

three octet identifier (tag) which identifies a specific piece of application information

3.4.26

guardband

time slot allocated for the transmission of the moderator DLPDU

3.4.27

implicit token

mechanism that governs the right to transmit

Note 1 to entry: No actual token message is transmitted on the medium. Each node keeps track of the MAC ID of the node that it believes currently holds the right to transmit. The right to transmit is passed from node to node by keeping a record of the node that last transmitted. A slot time is used to allow a missing node to be skipped in the rotation

3.4.28

implicit token register

register that contains the MAC ID of the node that holds the right to transmit

3.4.29

instance

actual physical occurrence of an object within a class, identifying one of many objects within the same object class

EXAMPLE California is an instance of the object class state.

Note 1 to entry: The terms object, instance, and object instance are used to refer to a specific instance.

3.4.30

instance attribute

attribute that is unique to an object instance and not shared by the object class

3.4.31

instantiated

object that has been created in a device

3.4.32

Keeper

object responsible for distributing link configuration data to all nodes on the link

3.4.33

link

collection of nodes with unique MAC IDs, attached to Ph-segments connected by Ph-repeaters

3.4.34

little endian

model of memory organization which stores the least significant octet at the lowest address, and of transmission organization which transfers the least significant octet first on the medium

3.4.35

Lpacket

well-defined sub-portion of a DLPDU consisting of

- a) size and control information
- b) a fixed tag or a generic tag, and
- c) DLS-user data or, when the tag has DL-significance, DL-data

3.4.36

M_symbol(s)

symbols that represent the data bits and related information encoded and transmitted by the PhL

3.4.37

maximum scheduled node

node with highest MAC ID that can use scheduled time on a link

3.4.38

maximum unscheduled node

node with highest MAC ID that can use unscheduled time on a link

3.4.39

Message Router

object within a node that distributes messaging requests to the appropriate application objects

3.4.40

moderator

node with the lowest MAC ID that is responsible for transmitting the moderator DLPDU

3.4.41

moderator DLPDU

DLPDU transmitted by the node with the lowest MAC ID for the purpose of synchronizing the nodes and distributing the link configuration parameters

3.4.42

multipoint connection

connection from one node to many

Note 1 to entry: Multipoint connections allow messages from a single producer (publisher) to be received by many consumer (subscriber) nodes.

3.4.43

network

series of nodes connected by some type of communication medium

Note 1 to entry: The connection paths between any pair of nodes can include repeaters, routers and gateways.

3.4.44

network access monitor

NAM

component within a node that manages communication with a temporary node connected to the nodes local NAP

3.4.45

network access port

NAP

PhL variant that allows a temporary node to be connected to the link by connection to the NAP of a permanent node

3.4.46

network address or node address

node's address on the link

Note 1 to entry: This is also called MAC ID.

3.4.47

network status indicators

indicators on a node indicating the status of the Physical and Data Link Layers

3.4.48

network update time

NUT

repetitive time interval in which data can be sent on the link

3.4.49

node

logical connection to a local link, requiring a single MAC ID

Note 1 to entry: A single physical device may appear as many nodes on the same local link. For the purposes of this protocol, each node is considered to be a separate DLE.

3.4.50

non-concurrence

situation where a transmission is received from an unexpected MAC ID, which appears to violate the time based access protocol

Note 1 to entry: This may occur when a connection is made between two working links that are not synchronized with each other but who have the same configuration information.

3.4.51

non-data symbol

PhL symbol which violates the requirements of Manchester biphase L encoding

3.4.52

object

abstract representation of a particular component within a device

Note 1 to entry: An object can be

- 1) an abstract representation of a computer's capabilities. Objects can be composed of any or all of the following components:
 - a) data (information which changes with time);
 - b) configuration (parameters for behavior);
 - c) methods (things that can be done using data and configuration);
- 2) a collection of related data (in the form of variables) and methods (procedures) for operating on that data that have clearly defined interface and behavior.

3.4.53

object specific service

service defined by a particular object class to perform a required function which is not performed by a common service

Note 1 to entry: An object specific service is unique to the object class which defines it.

3.4.54

originator

client responsible for establishing a connection path to the target

3.4.55

permanent node

node whose connection to the network does not utilize the network access port (NAP) PhL variant

Note 1 to entry: This node may optionally support a NAP PhL variant to allow temporary nodes to connect to the network.

3.4.56

produce

act of sending data to be received by a consumer

3.4.57

producer

node that is responsible for sending data

3.4.58

redundant media

system using more than one medium to help prevent communication failures

3.4.59

requested packet interval

RPI

measure of how frequently the originating application requires the transmission of Lpackets or data packets from the target application

3.4.60

rogue

node that has received a moderator DLPDU that disagrees with the link configuration currently used by this node

3.4.61

scheduled

data transfers that occur in a deterministic and repeatable manner on predefined NUTs

3.4.62

server

object which provides services to another (client) object

3.4.63

service

operation or function than an object and/or object class performs upon request from another object and/or object class

Note 1 to entry: A set of common services is defined and provisions for the definition of object-specific services are provided. Object-specific services are those which are defined by a particular object class to perform a required function which is not performed by a common service.

3.4.64

slot time

maximum time required for detecting an expected transmission

Note 1 to entry: Each node waits a slot time for each missing node during the implied token pass.

3.4.65

start delimiter

unique set of M_symbols that identifies the beginning of a DLPDU

3.4.66

supernode

DLE with a MAC ID of zero

Note 1 to entry: This DLE DL-address is reserved for special DLL functions

3.4.67

table unique identifier

TUI

unique reference code used by ControlNet and Keeper objects to reference a consistent set of link configuration attribute

3.4.68

tag

shorthand name for a specific piece of application information, two or three octets in length

3.4.69

target

end-node to which a connection is established

3.4.70

temporary node

same as transient node

3.4.71

tMinus

number of NUTs before a new set of link configuration parameters are to be used

3.4.72

tone

instant of time which marks the boundary between two NUTs

3.4.73

tool

executable software program which interacts with the user to perform some function

EXAMPLE Link scheduling software (LSS).

3.4.74

transaction id

field within a UCMM header that matches a response with the associated request, which a server echoes in the response message

3.4.75

transient node

node that is only intended to be connected to the network on a temporary basis using the NAP PhL medium connected to the NAP of a permanent node

3.4.76

Unconnected Message Manager UCMM

component within a node that transmits and receives unconnected explicit messages and sends them directly to the Message Router object

3.4.77

unconnected service

messaging service which does not rely on the set up of a connection between devices before allowing information exchanges

3.4.78

unscheduled

data transfers that use the remaining allocated time in the NUT after the scheduled transfers have been completed

3.4.79

vendor ID

identification of each product manufacturer/vendor by a unique number

Note 1 to entry: Vendor IDs are assigned by ODVA, Inc.

3.5 Type 2 symbols and abbreviations

ACD	Address Conflict Detection	
ACM	Access control machine	
BOOTP	Bootstrap Protocol	[IETF RFC 951]
CA	Clock accuracy	
CAN	Controller area network	[ISO 11898:1993]
COP	Connection originator password	
DHCP	Dynamic Host Configuration Protocol	[IETF RFC 2131]
DLR	Device level ring	
DSCP	DiffServ code point	
FCS	Frame Check Sequence	
IP	Internet Protocol	[IETF RFC 791]
LED	Light emitting diode.	
LSS	Link scheduling software	
MAC ID	MAC address of a node	
MSTP	Multiple spanning tree protocol	[IEEE 802.1Q]

NAM	Network attachment monitor	
NAP	Network access port	
ND	Non-data symbol	
NUT	Network update time	
PT	Programming terminal (a temporary network connection)	
QoS	Quality of service	
Rcv	Receive	
RPI	Requested Lpacket interval	
RSTP	Rapid spanning tree protocol	[IEEE 802.1D]
Rx	Receive	
RxLLC	Receive logical link control	
RxM	Receive machine	
SEM	State event matrix	
SMAX	MAC ID of the maximum scheduled node	
STD	State transition diagram, used to describe object behavior	
STP	Spanning tree protocol	[IEEE 802.1D]
TCP	Terminal Control Protocol	[IETF RFC 793]
TFTP	Trivial File Transfer Protocol	[IETF RFC 1350]
TUI	Table unique identifier	
Tx	Transmit	
TxLLC	Transmit logical link control	
TxM	Transmit machine	
UCCM	Unconnected Message Manager	
UMAX	MAC ID of maximum unscheduled node	
USR	Unscheduled start register	

3.6 Conventions for station management objects

Objects for station management are specified in Clause 7, using attributes, services and behaviour.

Object attributes are specified in tables formatted according to Table 1.

Table 1 – Format of attribute tables

Attribute ID	Need in implementation	Access rule	NV	Name	Data type	Description of attribute	Semantics of values
See a)	See b)	See c)	See d)	See e)	See 6.1.3.	See f)	See g)

- a) "Attribute ID" is the identification value assigned to an attribute.
- b) "Need in implementation" specifies whether or not the attribute is necessary in the implementation. An attribute may be optional, required or conditional.
- A conditional attribute is required if certain object behaviours and/or attributes are implemented as defined by the class.

If an Attribute is optional, then a default value shall be defined in the semantics. An optional attribute's default value shall provide the same behaviour as if the attribute was not implemented. If the default value of an optional attribute does not match the behaviour of the object when the object does not implement the attribute, the object definition shall declare the behaviour.

In all cases, the term “default” indicates a “factory default” as shipped from the vendor.

- c) “Access rule” specifies how a requestor can access an attribute. The definitions for access rules are:
 - Settable (Set) – The attribute shall be accessed by at least one of the set services. If the behaviour of the device does not require a set service, then a device implementation of that object is not required to implement the attribute as settable.
Settable attributes, unless otherwise specified by the object definition, shall be also accessed by get services.
 - Gettable (Get) – The attribute shall be accessed by at least one of the get services.
- d) “NV” indicates whether an attribute value is maintained through power cycles. This column is used in object definitions where non-volatile storage of attribute values is required. An entry of “NV” indicates value shall be saved, “V” means not saved.
- e) Name refers to the attribute.
- f) Description of Attribute provides general information about the attribute.
- g) Semantics of Values specifies the meaning of the value of the attribute.

4 Overview of the data-link protocol

4.1 General

4.1.1 DLL architecture

The Type 2 DLL is modeled as an integrated Access Control Machine (ACM), with scheduling support functions, designed for reliable and efficient support of higher-level connection-mode and connectionless data transfer services. Interoperating with these higher level functions are DLL management functions.

PhL framing and delimiters are managed by DLL functions for serializing and deserializing M_symbol requests and indications.

Within the Data Link Layer, the Access Control Machine (ACM) has the primary responsibility for detecting and recovering from network disruptions. The major objectives of the ACM are

- to make sure that the local node detects and fully utilizes its assigned access slots in the schedule;
- to make sure that the local node does not interfere with the transmissions of other nodes, especially the moderator node;
- to start the next NUT (see 4.1.2) on schedule, whether the moderator DLPDU is heard or not;
- if the local node is the moderator, to transmit each moderator DLPDU strictly on schedule.

The Data Link Layer is comprised of the components listed in Table 2:

Table 2 – Data-link layer components

Components	Description
Access Control Machine (ACM)	receives and transmits control DLPDUs and header information, and determines the timing and duration of transmissions.
Transmit LLC (TxLLC)	buffers DLSDUs received from Station Management and the DLS-user, and determines which should be transmitted next.
Receive LLC (RxLLC)	performs the task of quarantining received link units until they are validated by a good FCS.
Transmit Machine (TxM)	receives requests to send DLPDU headers and trailers, and Lpackets from the ACM, and breaks them down into octet symbol requests that are sent to the Serializer.
Receive Machine (RxM)	assembles received Lpackets from octet symbols received from the Deserializer, and submits them to the RxLLC.
Serializer	receives octet symbols, encodes and serializes them, and sends them as M_symbols to the PhL. It is also responsible for generating the FCS.
Deserializer	receives M_symbols from the PhL, converts M_symbols into octets and sends them to the receive machine. It is also responsible for checking the FCS.
DLL Management Interface	holds the Station Management variables that belong to the DLL, and helps manage synchronized changes of the link parameters.

The internal arrangement of these components, and their interfaces, are shown in Figure 2. The arrowheads illustrate the primary direction of flow of data and control.

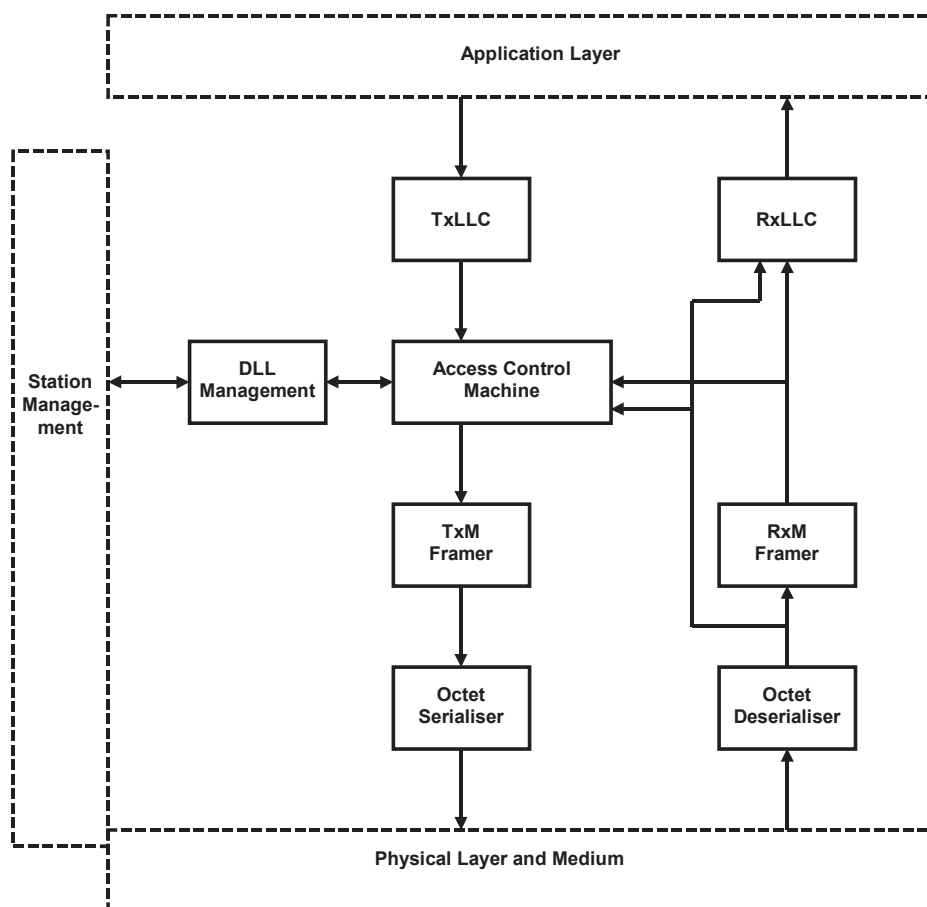


Figure 2 – Data-link layer internal architecture

4.1.2 Access control machine (ACM) and scheduling support functions

The ACM functions schedule all communication from one DLE to another. The timing of this communication are regulated at two levels,

- 1) to provide an accurate time based distribution of scheduled opportunities for designated communications in accord with, and with a pre-established schedule for, the local link and any extended network of links,
- 2) to provide a democratic distribution of unscheduled opportunities for arbitrary communications, generally in a cyclic but asynchronous manner.

Accurate schedule timing as provided by 1) is very important to support many control and data collection tasks in the applications domain of this protocol. The schedule is based on a fixed, repetitive time cycle called network update time (NUT). The NUT is maintained in close synchronism among all nodes participating in the DLL and used to provide two types of access opportunity;

- 1) One opportunity for each active station to transfer one scheduled DLPDU containing multiple scheduled DLSDUs.

Only DLSDUs with scheduled as their QoS parameter may be included in a scheduled DLPDU.

NOTE DLSDUs with scheduled as their QoS parameter are usually connection mode transfers with generic as their tag parameter.

- 2) One opportunity for at least one active station to send one background (unscheduled) DLPDU containing multiple DLSDUs of any QoS type. If time is available within a NUT, one background MAC opportunity is offered to each other active station in order of their MAC ID address. The station MAC ID for the first background opportunity in each NUT, is incremented (+1 modulo the set of background addresses) for each successive NUT to democratically share the available time among active stations.

Support procedures are included for automatic transfer to backup scheduling services and to manage the use of redundant Ph media.

4.1.3 Connection-mode, connectionless-mode data transfer and DL service

The connection mode, connectionless mode and DL services provide the necessary functionality for

- managing DL provider interactions with the DL user by converting request and response primitives into necessary DLE operations and generating indications and confirm primitives where appropriate,
- determination of the DL address and control information to be added to each DLSDU,
- local confirmation of status for each outgoing DLSDU,
- DLPDU formation by concatenation of multiple DL user DLSDUs for efficient transfer as single PhPDUs, subject to DLPDU size limitations, QoS parameters and the type of access opportunity.

4.2 Services provided by the DL

4.2.1 Overview

The DLL provides connectionless data transfer services and connection-mode data transfer services for limited-size DLSDUs, an internally synchronized time service, scheduling services to control the time allocation of the underlying shared PhL service, a DL(SAP)-address and queue management service, and a packing/unpacking service that combines multiple DLSDUs into a single data transfer DLPDU for efficient use of each access opportunity.

4.2.2 QoS

4.2.2.1 Definition

The QoS parameter specifies priority options for DLSDUs and access opportunities. The available values are as follows:

4.2.2.2 DLL scheduled priority

The scheduled priority QoS provides accurate time based cyclic and acyclic sending of DLSDUs. The execution timing for this fieldbus scheduled service can be accurate and repeatable to better than 1 ms. Scheduled priority is normally used with connected mode services.

NOTE The master Keeper is responsible for regular publication of a unique table identifier (TUI) which ensures all nodes have a reference for the current set of link operating parameters. This TUI publication uses a fixed tag and is sent with QoS priority set to scheduled.

4.2.2.3 DLL high priority (unscheduled)

The high priority QoS provides acyclic sending of DLSDUs with a bounded upper time for the sending delay. Data on this priority is sent only when all scheduled data has been sent and a non-scheduled sending opportunity is available.

4.2.2.4 DLL low priority (unscheduled)

The low priority QoS provides for sending of DLSDUs only on an as-available basis. Data on this priority is sent only when all other data priorities have been sent and a non-scheduled sending opportunity is available.

NOTE High and Low priority are used only in a local sense to set the order of servicing among locally submitted, unscheduled DLS-user-data.

4.3 Structure and definition of DL-addresses

4.3.1 General

DL-addresses are used as MAC ID addresses (Link node designators), Fixed tag addresses (DLSAP-addresses and group DL-addresses), and Generic tag addresses (DLCEP-addresses).

Subclause 4.3 defines the form and structure of DL-addresses and includes specific definitions for some standard DL-addresses.

All addresses are formed of octets. Each octet shall be transferred to the Ph layer low-bit first and multiple octets shall be transferred low-order octet first (little endian format).

Three types of DL addresses are used.

4.3.2 MAC ID address

MAC ID addresses identify physical nodes on the local link as shown in Figure 3.

Permitted values are within the range 0 to 99, values from 100 to 254 are reserved.



Figure 3 – Basic structure of a MAC ID address

Table 3 specifies the use of MAC ID addresses.

Table 3 – MAC ID addresses allocation

MAC ID	Meaning and purpose
0x0	temporary address used for maintenance
SMAX	current maximum address value for Scheduled access opportunities. SMAX =< UMAX
UMAX	current maximum address value for Unscheduled access opportunities SMAX. =< UMAX =< 0x63
0x64 to 0xFE	reserved
0xFF	broadcast address to all MAC ID nodes on this link

4.3.3 Generic tag address

Generic tag addresses types identify all data transfers for connected mode services. The available capacity in the DLL is 3 octets, see Figure 4.

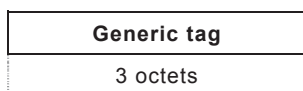


Figure 4 – Basic structure of a generic tag address

Generic tags are used to send connected DLSDUs and may be associated with any of the available QoS priorities (Scheduled, High, Low).

NOTE Negotiation and management of generic tags and connected mode services are the responsibility of the DL user.

4.3.4 Fixed tag address

Fixed tag addresses types identify DLSAPs within a designated station MAC ID. They are used with all data transfers for connectionless mode services.

NOTE Fixed tags are used to send connectionless DLSDUs and are usually associated with unscheduled QoS priorities (High, Low). Typical uses of fixed tags are for station management and to establish connections.

Each fixed tag address shall consist of two octets as shown in Figure 5.

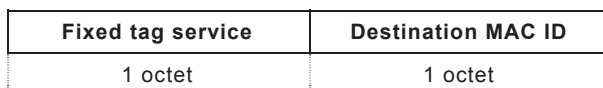


Figure 5 – Basic structure of a fixed tag address

The first octet shall specify the service access point in the destination node as specified in Table 4. The second octet shall be the address of the destination node. The destination address shall be either a MAC ID or 0xFF, which indicates the broadcast address to all MAC IDs on the local link.

Table 4 – Fixed tag service definitions

Fixed tag service	Meaning
0x00	moderator
0x01 - 0x08	vendor specific
0x09	ping request
0x0A – 0x14	vendor specific
0x15	tMinus
0x16 - 0x28	vendor specific
0x29	ping reply
0x2A - 0x3F	vendor specific
0x40 - 0x6F	reserved
0x70 – 0x7F	vendor-specific
0x80	I'm alive
0x81	link parameters
0x82	reserved
0x83	UCMM
0x84	TUI
0x85	IP (reserved for Internet Protocol)
0x86	WAMI
0x87	reserved
0x88	Keeper UCMM
0x89	Ethernet (used for address resolution protocol)
0x8A - 0x8B	reserved
0x8C	Time distribution
0x8D – 0x8F	reserved for time distribution
0x90	debug
0x91 - 0xCF	reserved
0xD0 - 0xEF	group addresses
0xF0 - 0xFF	vendor specific

Fixed tags in the range 0xD0 to 0xEF shall be reserved to permit group screening of connection identifiers.

4.4 Services assumed from the PhL

4.4.1 General requirements

Subclauses 4.4.2 to 4.4.3 include requirements for interface to a Type 2 PhL that are not required for other Types.

- Transmit Machine.
- Receive Machine.
- Serializer.
- Deserializer.

This is necessary because the Ph-Interface Data Units (PhIDUs) that Type 2 passes across the DLL–PhL interface are individual serial data and non-data symbols, designated on the DLE side as M_Symbols (see Table 5), whereas PhIDUs for other Types are not conceptualized in this way. As a consequence, a Type 2 Data Link Layer requires a Type 2 PhL.