

DIN EN 61158-4-2

ICS 35.100.20; 35.160

Ersatz für
DIN EN 61158-4-2:2008-09
Siehe Anwendungsbeginn

**Industrielle Kommunikationsnetze –
Feldbusse –
Teil 4-2: Protokollspezifikation des Data Link Layer
(Sicherheitsschicht) –
Typ 2-Elemente
(IEC 61158-4-2:2010);
Englische Fassung EN 61158-4-2:2012, nur auf CD-ROM**

Industrial communication networks –
Fieldbus specifications –
Part 4-2: Data-link layer protocol specification –
Type 2 elements
(IEC 61158-4-2:2010);
English version EN 61158-4-2:2012, only on CD-ROM

Réseaux de communication industriels –
Spécifications de bus de terrain –
Partie 4-2: Spécification du protocole de couche de liaison de données –
Éléments de Type 2
(CEI 61158-4-2:2010);
Version anglaise EN 61158-4-2:2012, seulement en CD-ROM

Gesamtumfang 288 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2012-03-28 angenommene Europäische Norm als DIN-Norm ist 2013-01-01.

Für DIN EN 61158-4-2:2008-09 gilt eine Übergangsfrist bis zum 2015-03-28.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN IEC 61158-400:2009-06.

Für dieses Dokument ist das nationale Arbeitsgremium K 956 „Feldbus“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik im 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 3, der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer 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 vorliegende Norm legt das Protokoll der Datensicherungsschicht von Typ 2 fest.

Erklärung zu Schutzrechten

Die Internationale Elektrotechnische Kommission macht darauf aufmerksam, dass ihr gegenüber erklärt wurde, dass die Übereinstimmung mit dem vorgesehenen Norm-Inhalt die Anwendung von Patenten erforderlich machen könnte. Nähere Informationen hierzu befinden sich in der „Introduction“.

Es wird darauf aufmerksam gemacht, dass Bestandteile der vorliegenden Norm Gegenstand von weiteren Patentrechten sein können. Es liegt nicht im Verantwortungsbereich der DKE, einige oder alle dieser Patentrechte zu benennen.

Der Leser der vorliegenden Norm wird jedoch gebeten, ihm bekannte derartige Patentrechte der Geschäftsstelle mitzuteilen und entsprechende Informationen zur Verfügung zu stellen.

Änderungen

Gegenüber DIN EN 61158-4-2:2008-09 wurden folgende Änderungen vorgenommen:

- a) Abschnitt 2 und die Literaturhinweise auf den neuesten Stand gebracht;
- b) in 3.5 die Abkürzungen auf den neusten Stand gebracht;
- c) in 7.6 das Ethernet-Link-Objekt auf den neuesten Stand gebracht;
- d) in 7.7 geringfügige Änderungen am Devicenet-Objekt (ein zusätzliches Attribut) vorgenommen;
- e) in 7.9 und 7.10 neue DLR- und QoS-Objekte eingeführt;
- f) einen neuen Abschnitt 10 mit einem neuen DLR-Protokoll eingeführt.

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

Nationaler Anhang NA (informativ)

3.3 Allgemeine 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

3.3.2

DLSAP

(en: **DLSAP**)

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

ANMERKUNG Diese von ISO/IEC 7498-1 abgeleitete Definition wird an dieser Stelle wiederholt, um den entscheidenden Unterschied zwischen DLSAPs und deren DL-Adressen besser verstehen zu können. (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 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 DLSAP innerhalb des erweiterten Verbindungszweiges kennzeichnet

ANMERKUNG Eine einzelne DL-Instanz 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-Instanzen geschaltet werden können, wobei ein einziger DL-Namensraum (DL-Adressraum) gemeinsam genutzt wird; in diesem Subnetzwerk können alle angeschlossenen DL-Instanzen miteinander entweder direkt oder mithilfe einer oder mehrerer der zwischengeschalteten DL-Relais-Instanzen kommunizieren

ANMERKUNG 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 Verbindungszweiges kennzeichnet. Eine einzelne DL-Instanz kann mehrere DL-Gruppenadressen, die einem einzelnen DLSAP zugeordnet sind, besitzen. Eine einzelne DL-Instanz kann auch eine einzelne DL-Gruppenadresse, die mehr als einem DLSAP zugeordnet ist, besitzen.

3.3.8

Knoten

(en: **node**)

einzelne DL-Instanz, 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 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

3.4 Zusätzliche Begriffe für Typ 2

3.4.1

tatsächliches Paketintervall

(en: **actual packet interval, API**)

Maß, wie häufig eine spezifische Verbindung ihre LPaketdaten erzeugt

3.4.2

zuweisen

(en: **allocate**)

eine Ressource aus einem allgemeinen Bereich für die ausschließliche Nutzung durch eine spezifische Einheit zuweisen

3.4.3

Anwendung

(en: **application**)

Funktion oder Datenstruktur, für die Daten verarbeitet oder erzeugt werden

3.4.4

Anwendungsobjekte

(en: **application objects**)

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

3.4.5

Attribut

(en: **attribute**)

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

ANMERKUNG Die Attribute eines Objekts enthalten Informationen zu veränderlichen Teilen eines Objekts. Üblicherweise stellen sie Statusinformationen bereit 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**)

Angabe, wie das Objekt auf bestimmte Ereignisse reagiert. 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. 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 Objekts (Servers) nutzt, um eine Aufgabe auszuführen

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 während der Laufzeit verwalten und bereitstellen, z. B. das ControlNet-Objekt Unconnected Message Manager (UCCM)

3.4.11

Verbindung

(en: **connection**)

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

3.4.12

Verbindungs-ID

(en: **connection ID, CID**)

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

ANMERKUNG Auf der Sicherungsschicht entspricht dies einem generischen Tag.

3.4.13

zyklische Redundanzprüfung

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

3.4.16

Gerät

(en: **device**)

physikalische Hardware-Verbindung zum Verbindungszweig

ANMERKUNG Ein Gerät kann mehr als einen Knoten umfassen.

3.4.17

DLPDU

(en: **data-link protocol data unit**)

Protokolldateneinheit auf der Sicherungsschicht

ANMERKUNG Eine DLPDU besteht aus einer MAC-ID, null oder mehr LPaketen und einer Blockprüfzeichenfolge, so 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: frame check sequence)**

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 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 ausgeführt werden soll

ANMERKUNG Die Identifikation des/der Zielknoten(s) ist in dem Tag aus zwei Oktetten enthalten.

3.4.23**Rahmen****(en: frame)**

umgangssprachliches Synonym für DLPDU

3.4.24**Blockprüfzeichenfolge****(en: frame check sequence, FCS)**

redundante Daten, die mittels Hash-Funktion aus einem Block von Daten innerhalb einer DLPDU (Rahmen) abgeleitet und zusammen mit dem Block von Daten zur Erkennung von Datenverfälschungen gespeichert oder übertragen werden

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)**

Zeitfenster für die Übertragung der Moderator-DLPDU

3.4.27**implizites Token****(en: implicit token)**

Mechanismus, der das Recht zur Übertragung steuert

ANMERKUNG 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 weitergegeben, wobei der Knoten, der zuletzt übertragen hat, aufgezeichnet wird. Damit ein fehlender Knoten im Kreislauf übersprungen werden kann, wird ein Zeitfenster 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 physikalisches Vorkommen eines Objekts innerhalb einer Klasse, das ein oder mehrere Objekte innerhalb derselben Objektklasse identifiziert

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

ANMERKUNG 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 von der Objektklasse nicht gemeinsam genutzt wird

3.4.31

instanziiert

(en: **instantiated**)

Objekt, das in einem Gerät erstellt wurde

3.4.32

Keeper

(en: **Keeper**)

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

3.4.33

Verbindungszweig

(en: **link**)

Gruppe 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

LPaket

(en: **Lpacket**)

klar abgegrenzter Unterabschnitt einer DLPDU, bestehend aus

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

3.4.36

M_Zeichen

(en: **M_symbol(s)**)

Symbole, die die Datenbits und zugehörigen Informationen darstellen, die von der Bitübertragungsschicht 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 Nachrichtenanforderungen 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 Verbindungszweiges übertragen wird

3.4.42**Mehrpunktverbindung****(en: multipoint connection)**

Verbindung von einem Knoten zu vielen Knoten

ANMERKUNG 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. 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, die die Kommunikation mit einem temporären Knoten, der mit den lokalen NAPs des Knotens verbunden ist, steuert

3.4.45**Netzwerkzugangspunkt****(en: network access port, NAP)**

Variante der Bitübertragungsschicht, durch die ein temporärer Knoten mit dem Verbindungszweig über eine Verbindung eines permanenten Knotens verbunden werden kann

3.4.46**Netzwerkadresse oder Knotenadresse****(en: network address or node address)**

Adresse eines Knotens auf dem Verbindungszweig (auch als MAC-ID bezeichnet)

3.4.47

Netzwerkstatusanzeigen

(en: **network status indicators**)

Anzeigen an einem Knoten, die den Status der Bitübertragungs- und der Sicherungsschicht anzeigen

3.4.48

NUT

(en: **network update time**)

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 Ein einzelnes physikalisches 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 Eine Nichtübereinstimmung kann auftreten, wenn eine Verbindung zwischen zwei betriebsfähigen Verbindungszweigen hergestellt wird, die nicht miteinander synchronisiert werden, aber über die gleichen Konfigurationsinformationen verfügen.

3.4.51

Nicht-Datensymbol

(en: **non-data symbol**)

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

3.4.52

Objekt

(en: **object**)

abstrakte Darstellung einer bestimmten Komponente innerhalb eines Gerätes

ANMERKUNG Ein Objekt kann sein:

- 1) eine abstrakte Darstellung der Fähigkeiten eines Computers. Objekte können aus einigen der folgenden Komponenten 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) einer Zusammenstellung zugehöriger Daten (in Form von Variablen) und Methoden (Prozeduren) für Operationen in Bezug auf diese Daten mit klar definierter Schnittstelle und klar definiertem Verhalten.

3.4.53

objektspezifischer Dienst

(en: **object specific service**)

Dienst, der durch eine bestimmte Objektklasse definiert ist um eine erforderliche Funktion, die nicht von einem allgemeinen Dienst ausgeführt wird, auszuführen

ANMERKUNG Ein objektspezifischer Dienst ist für die Objektklasse, die ihn definiert, eindeutig.

3.4.54

Absender

(en: **originator**)

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

3.4.55

permanenter Knoten

(en: **permanent node**)

Knoten, dessen Verbindung zum Netzwerk keinen Netzwerkzugangspunkt (NAP) der Variante der Bitübertragungsschicht verwendet

ANMERKUNG Dieser Knoten kann optional eine NAP Bitübertragungsschichtvariante unterstützen, um die Verbindung temporärer Knoten mit dem Netzwerk zu ermöglichen.

3.4.56

erzeugen

(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

(en: **requested packet interval, RPI**)

Maß, wie häufig die absendende Anwendung die Übertragung von LPaketen 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 Verbindungszweiges, die dieser Knoten kürzlich verwendet hat, übereinstimmt

3.4.61

geplant

(en: **scheduled**)

Datentransfers, die plangesteuert und wiederholt zu vorgegebenen NUTs auftreten

3.4.62

Server

(en: **server**)

Objekt, das einem anderen Objekt (Client) Dienste bereitstellt

3.4.63

Dienst

(en: **service**)

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

ANMERKUNG Es ist eine Reihe von allgemeinen Diensten definiert, und es sind Richtlinien für die Definition von objektspezifischen Diensten angegeben. Objektspezifische Dienste sind durch eine bestimmte Objektklasse definierte Dienste, die eine erforderliche Funktion, die nicht von einem allgemeinen Dienst ausgeführt wird, ausführen sollen.

3.4.64

Zeitfenster

(en: slot time)

erforderliche maximale Dauer für die Erkennung einer erwarteten Übertragung

ANMERKUNG Während des implizierten Tokendurchlaufs wartet jeder Knoten für ein Zeitfenster auf jeden fehlenden Knoten.

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 Die DL-Adresse dieser DLE ist für spezielle DLL-Funktionen reserviert.

3.4.67

eindeutige Tabellenkennung

(en: table unique identifier, TUI)

eindeutiger Referenzcode, der von ControlNet- und Keeper-Objekten für die Referenzierung einer konsistenten Menge von Konfigurationsattributen des Verbindungszweiges 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)

das Gleiche wie transienter 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 Verbindungszweiges.

3.4.74

Transaktions-ID

(en: **transaction id**)

Feld innerhalb eines UCMM-Headers, das ein Server in der Antwortnachricht wiederholt, um eine Antwort der zugehörigen Anforderung zuzuordnen

3.4.75

transienter Knoten

(en: **transient node**)

Knoten, der nur temporär mithilfe des NAP Bitübertragungsschichtmediums, 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 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 61158-4-2:2010)**

Réseaux de communication industriels -
Spécifications de bus de terrain -
Partie 4-2: Spécification du protocole de
couche de liaison de données -
Eléments de Type 2
(CEI 61158-4-2:2010)

Industrielle Kommunikationsnetze -
Feldbusse -
Teil 4-2: Protokollspezifikation des Data
Link Layer (Sicherungsschicht) -
Typ 2-Elemente
(IEC 61158-4-2:2010)

This European Standard was approved by CENELEC on 2012-03-28. 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, 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.

CENELEC

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

Management Centre: Avenue Marnix 17, B - 1000 Brussels

Foreword

The text of document 65C/605/FDIS, future edition 2 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 61158-4-2:2012.

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) 2012-12-28
- latest date by which the national standards conflicting with the document have to be withdrawn (dow) 2015-03-28

This document supersedes EN 61158-4-2:2008.

EN 61158-4-2:2012 includes the following significant technical changes with respect to EN 61158-4-2:2008:

- Clause 2 and Bibliography: update of normative and bibliographic references;
- 3.5: update of abbreviations;
- 7.6: updates of the Ethernet Link object;
- 7.7: minor update of the Devicenet object (additional attribute);
- new 7.9 and 7.10: new DLR and QoS objects;
- new Clause 10: new DLR protocol.

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.

Endorsement notice

The text of the International Standard IEC 61158-4-2:2010 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/TR 61158-1:2010 | NOTE Harmonized as CLC/TR 61158-1:2010 (not modified). |
| IEC 61158-2:2010 | NOTE Harmonized as EN 61158-2:2010 (not modified). |

CONTENTS

INTRODUCTION	10
1 Scope	12
1.1 General.....	12
1.2 Specifications.....	12
1.3 Procedures	12
1.4 Applicability.....	13
1.5 Conformance	13
2 Normative references	13
3 Terms, definitions, symbols and abbreviations.....	14
3.1 Reference model terms and definitions.....	14
3.2 Service convention terms and definitions.....	16
3.3 Common terms and definitions	17
3.4 Additional Type 2 definitions.....	18
3.5 Type 2 symbols and abbreviations.....	25
4 Overview of the data-link protocol.....	26
4.1 General.....	26
4.2 Services provided by the DL.....	29
4.3 Structure and definition of DL-addresses	29
4.4 Services assumed from the PhL	31
4.5 Functional classes.....	34
5 General structure and encoding of PhIDUs and DLPDUs and related elements of procedure.....	34
5.1 Overview.....	34
5.2 Media access procedure	34
5.3 DLPDU structure and encoding	38
5.4 Lpacket components	41
5.5 DLPDU procedures	43
5.6 Summary of DLL support services and objects	45
6 Specific DLPDU structure, encoding and procedures.....	46
6.1 Modeling language.....	46
6.2 DLS user services	48
6.3 Generic tag Lpacket.....	54
6.4 Moderator Lpacket	54
6.5 Time distribution Lpacket.....	56
6.6 UCMM Lpacket	58
6.7 Keeper UCMM Lpacket.....	58
6.8 TUI Lpacket	59
6.9 Link parameters Lpacket and tMinus Lpacket	60
6.10 I'm-alive Lpacket	62
6.11 Ping Lpackets	63
6.12 WAMI Lpacket.....	64
6.13 Debug Lpacket.....	65
6.14 IP Lpacket.....	66
6.15 Ethernet Lpacket.....	66
7 Objects for station management.....	66
7.1 General.....	66

7.2	ControlNet object	67
7.3	Keeper object.....	76
7.4	Scheduling object.....	97
7.5	TCP/IP Interface object	107
7.6	Ethernet link object.....	116
7.7	DeviceNet object.....	122
7.8	Connection configuration object (CCO)	130
7.9	DLR object.....	148
7.10	QoS object.....	155
8	Other DLE elements of procedure	158
8.1	Network attachment monitor (NAM).....	158
8.2	Calculating link parameters	165
9	Detailed specification of DL components	173
9.1	General.....	173
9.2	Access control machine (ACM)	173
9.3	TxLLC.....	192
9.4	RxLLC	197
9.5	Transmit machine (TxM)	201
9.6	Receive machine (RxM).....	204
9.7	Serializer.....	211
9.8	Deserializer.....	213
9.9	DLL management	213
10	Device Level Ring (DLR) protocol	216
10.1	General.....	216
10.2	Supported topologies	216
10.3	Overview of DLR operation	217
10.4	Classes of DLR implementation	220
10.5	DLR behavior.....	221
10.6	Implementation requirements.....	226
10.7	Using non-DLR nodes in the ring network.....	228
10.8	DLR messages	231
10.9	State diagrams and state-event-action matrices	235
10.10	Performance analysis.....	254
	Annex A (normative) Indicators and switches	259
	Bibliography.....	272
	Annex ZA (normative) Normative references to international publications with their corresponding European publications.....	273
	Figure 1 – Relationships of DLSAPs, DLSAP-addresses and group DL-addresses.....	17
	Figure 2 – Data-link layer internal architecture.....	28
	Figure 3 – Basic structure of a MAC ID address	30
	Figure 4 – Basic structure of a generic tag address	30
	Figure 5 – Basic structure of a fixed tag address	31
	Figure 6 – M_symbols and Manchester encoding at 5 MHz.....	32
	Figure 7 – NUT structure.....	35
	Figure 8 – Media access during scheduled time	36
	Figure 9 – Media access during unscheduled time	37
	Figure 10 – DLPDU format.....	38

Figure 11 – Aborting a DLPDU during transmission.....	41
Figure 12 – Lpacket format	42
Figure 13 – Generic tag Lpacket format.....	43
Figure 14 – Fixed tag Lpacket format	43
Figure 15 – Goodness parameter of TimeDist_Lpacket	56
Figure 16 – Example I'm alive processing algorithm	63
Figure 17 – Keeper CRC algorithm.....	82
Figure 18 – Keeper object power-up state diagram.....	93
Figure 19 – Keeper object operating state diagram.....	94
Figure 20 – Synchronized network change processing	97
Figure 21 – State transition diagram for TCP/IP Interface object.....	115
Figure 22 – Connection configuration object edit flowchart	147
Figure 23 – NAM state machine	159
Figure 24 – DLR rings connected to switches.....	217
Figure 25 – Normal operation of a DLR network	218
Figure 26 – Beacon and Announce frames.....	218
Figure 27 – Link failure.....	219
Figure 28 – Network reconfiguration after link failure	220
Figure 29 – Neighbor Check process	226
Figure 30 – Unsupported topology – example 1.....	229
Figure 31 – Unsupported topology – example 2.....	229
Figure 32 – State transition diagram for Beacon frame based non-supervisor ring node	235
Figure 33 – State transition diagram for Announce frame based non-supervisor ring node	240
Figure 34 – State transition diagram for ring supervisor.....	243
Figure A.1 – Non redundant network status indicator labeling.....	263
Figure A.2 – Redundant network status indicator labeling.....	264
Table 1 – Data-link layer components	27
Table 2 – MAC ID addresses allocation.....	30
Table 3 – Fixed tag service definitions.....	31
Table 4 – Data encoding rules.....	32
Table 5 – M Data symbols.....	33
Table 6 – Truth table for ph_status_indication	33
Table 7 – FCS length, polynomials and constants	39
Table 8 – DLL support services and objects	45
Table 9 – Elementary data types.....	48
Table 10 – DLL events	52
Table 11 – Time distribution priority.....	57
Table 12 – Format of the TUI Lpacket	60
Table 13 – ControlNet object class attributes.....	67
Table 14 – ControlNet object instance attributes.....	68
Table 15 – TUI status flag bits.....	71
Table 16 – Channel state bits.....	72

Table 17 – ControlNet object common services.....	74
Table 18 – ControlNet object class specific services	75
Table 19 – Keeper object revision history.....	77
Table 20 – Keeper object class attributes.....	77
Table 21 – Keeper object instance attributes.....	77
Table 22 – Keeper operating state definitions.....	80
Table 23 – Port status flag bit definitions	80
Table 24 – TUI status flag bits.....	81
Table 25 – Keeper attributes	83
Table 26 – Memory requirements (in octets) for the Keeper attributes	84
Table 27 – Keeper object common services	84
Table 28 – Keeper object class specific services.....	85
Table 29 – Service error codes	86
Table 30 – Wire order format of the TUI Lpacket.....	90
Table 31 – Service error codes	90
Table 32 – Keeper object operating states	91
Table 33 – Keeper object state event matrix.....	95
Table 34 – Scheduling object class attributes.....	98
Table 35 – Scheduling object instance attributes.....	98
Table 36 – Scheduling object common services.....	99
Table 37 – Status error descriptions for Create	100
Table 38 – Status error descriptions for Delete and Kick_Timer.....	100
Table 39 – Scheduling object class specific services.....	101
Table 40 – Status error descriptions for Read.....	102
Table 41 – Status error descriptions for Conditional_Write	103
Table 42 – Status error descriptions for Forced_Write.....	104
Table 43 – Status error descriptions for Change_Start	104
Table 44 – Status error descriptions for Break_Connections.....	105
Table 45 – Status error descriptions for Change_Complete.....	106
Table 46 – Status error descriptions for Restart_Connections	106
Table 47 – TCP/IP Interface object class attributes	108
Table 48 – TCP/IP Interface object instance attributes	108
Table 49 – Status bits	110
Table 50 – Configuration capability bits.....	110
Table 51 – Configuration control bits.....	111
Table 52 – Example path	111
Table 53 – Interface configuration components	112
Table 54 – Alloc control values.....	113
Table 55 – TCP/IP Interface object common services.....	114
Table 56 – Get_Attribute_All reply format.....	114
Table 57 – Ethernet link object revision history.....	116
Table 58 – Ethernet link object class attributes.....	116
Table 59 – Ethernet link object instance attributes.....	117

Table 60 – Interface flags bits	119
Table 61 – Control bits	120
Table 62 – Interface type.....	121
Table 63 – Interface state.....	121
Table 64 – Admin state	121
Table 65 – Ethernet Link object common services	122
Table 66 – Ethernet Link object class specific services.....	122
Table 67 – DeviceNet object revision history	123
Table 68 – DeviceNet object class attributes	123
Table 69 – DeviceNet object instance attributes	123
Table 70 – Bit rate attribute values	125
Table 71 – BOI attribute values	126
Table 72 – Diagnostic counters bit description.....	127
Table 73 – DeviceNet object common services	128
Table 74 – Reset service parameter.....	129
Table 75 – Reset service parameter values.....	129
Table 76 – DeviceNet object class specific services	129
Table 77 – Connection configuration object revision history.....	130
Table 78 – Connection configuration object class attributes.....	131
Table 79 – Format number values.....	132
Table 80 – Connection configuration object instance attributes.....	133
Table 81 – Originator connection status values	135
Table 82 – Target connection status values.....	135
Table 83 – Connection flags.....	135
Table 84 – I/O mapping formats	137
Table 85 – Connection configuration object common services	138
Table 86 – Get_Attribute_All error codes.....	138
Table 87 – Get_Attribute_All response	139
Table 88 – Set_Attribute_All error codes	140
Table 89 – Set_Attribute_All request	140
Table 90 – Create request parameters	142
Table 91 – Create error codes.....	142
Table 92 – Delete error codes	142
Table 93 – Restore error codes.....	143
Table 94 – Connection configuration object class specific services.....	143
Table 95 – Kick_Timer error codes.....	143
Table 96 – Open_Connection error codes	144
Table 97 – Close_Connection error codes.....	144
Table 98 – Stop_Connection error codes	144
Table 99 – Change_Start error codes.....	145
Table 100 – Get_Status service parameter.....	145
Table 101 – Get_Status service response	145
Table 102 – Get_Status service error codes.....	146

Table 103 – Change_Complete service parameter	146
Table 104 – Change_Complete service error codes	146
Table 105 – Audit_Changes service parameter	147
Table 106 – Audit_Changes service error codes	147
Table 107 – Class attributes.....	148
Table 108 –Instance attributes	148
Table 109 – Network Status values	150
Table 110 – Ring Supervisor Status values.....	151
Table 111 – DLR object common services	154
Table 112 – Get_Attributes_All Response – non supervisor device	154
Table 113 – Get_Attributes_All Response – supervisor-capable device.....	154
Table 114 – DLR object class specific services	155
Table 115 – QoS object revision history	156
Table 116 – QoS object class attributes	156
Table 117 – QoS object instance attributes	156
Table 118 – Default DCSP values and usages	157
Table 119 – Common services.....	157
Table 120 – NAM states.....	158
Table 121 – Default link parameters.....	159
Table 122 – PhL timing characteristics	166
Table 123 – DLR variables.....	221
Table 124 – MAC addresses for DLR messages	231
Table 125 – IEEE 802.1Q frame format.....	231
Table 126 – Common fields in DLR messages	232
Table 127 – Format of the Beacon frame	232
Table 128 – Ring State values	233
Table 129 – Format of the Neighbor_Check request	233
Table 130 – Format of the Neighbor_Check response.....	233
Table 131 – Format of the Link_Status/Neighbor_Status frame	234
Table 132 – Link/Neighbor status values.....	234
Table 133 – Format of the Announce frame	234
Table 134 – Format of the Sign_On frame.....	235
Table 135 – Parameter values for Beacon frame based non-supervisor ring node.....	235
Table 136 – State-event-action matrix for Beacon frame based non-supervisor ring node	236
Table 137 – Parameter values for Announce frame based non-supervisor ring node	240
Table 138 – State-event-action matrix for Announce frame based non-supervisor ring node.....	241
Table 139 – Parameter values for ring supervisor node.....	244
Table 140 – State-event-action matrix for ring supervisor node	245
Table 141 – Parameters/assumptions for example performance calculations	255
Table 142 – Example ring configuration parameters and performance.....	258
Table A.1 – Module status indicator	260
Table A.2 – Network status indicators	261
Table A.3 – Network status indicator	265

Table A.4 – Network status indicator	267
Table A.5 – Combined module/network status indicator.....	268
Table A.6 – I/O status indicator	269
Table A.7 – Bit rate switch encoding.....	271

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/TR 61158-1:2010.

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.

NOTE Use of some of the associated protocol types is restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a particular data-link layer protocol type to be used with physical layer and application layer protocols in Type combinations as specified explicitly in the IEC 61784 series. Use of the various protocol types in other combinations may require permission from their respective intellectual-property-right holders.

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 12/493,838	[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 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://www.iec.ch/tctools/patent_decl.htm) maintain on-line data bases of patents relevant to their standards. Users are encouraged to consult the data bases 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 224 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 referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

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

IEC 61158-3-2:2007, *Industrial communication networks – Fieldbus specifications – Part 3-2: Data-link layer service definition – Type 2 elements*

IEC 61158-5-2:2010¹, *Industrial communication networks – Fieldbus specifications – Part 5-2: Application layer service definition – Type 2 elements*

IEC 61158-6-2:2010¹, *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 3309, *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*

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*

¹ To be published.

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

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

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

IETF RFC 1213, *Management Information Base for Network Management of TCP/IP-based internets : MIB-I*, 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>>

3 Terms, definitions, symbols and abbreviations

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]
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]

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

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

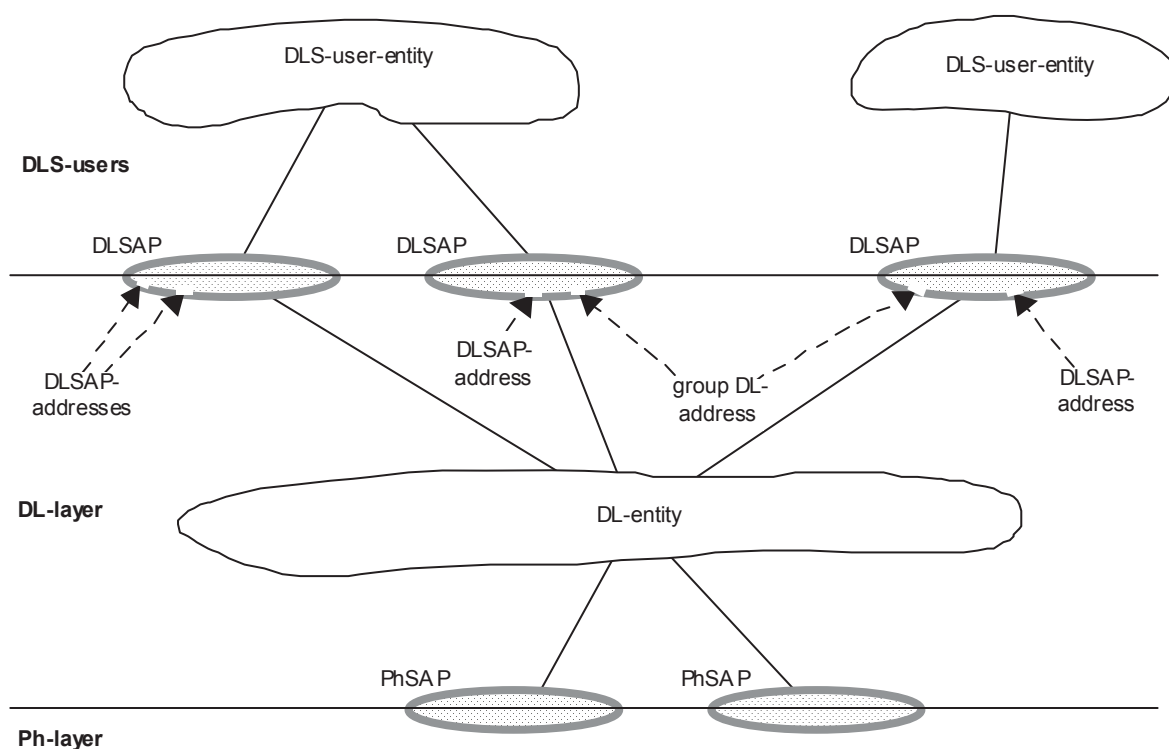
3.3 Common terms and definitions

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 may 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 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 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 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 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 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 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. Its description includes the relationship between attribute values and services

3.4.7

bit

unit of information consisting of a 1 or a 0. 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 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 A device may contain more than one node.

3.4.17

DLPDU

data-link protocol data unit

NOTE 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 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 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 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 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. 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 (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 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 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 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 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 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 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 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 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 Vendor IDs are assigned by ODVA, Inc.

3.5 Type 2 symbols and abbreviations

ACM	Access control machine	
CA	Clock accuracy	
CAN	Controller area network	[ISO 11898:1993]
COP	Connection originator password	
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]
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	

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 1 :

Table 1 – 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.

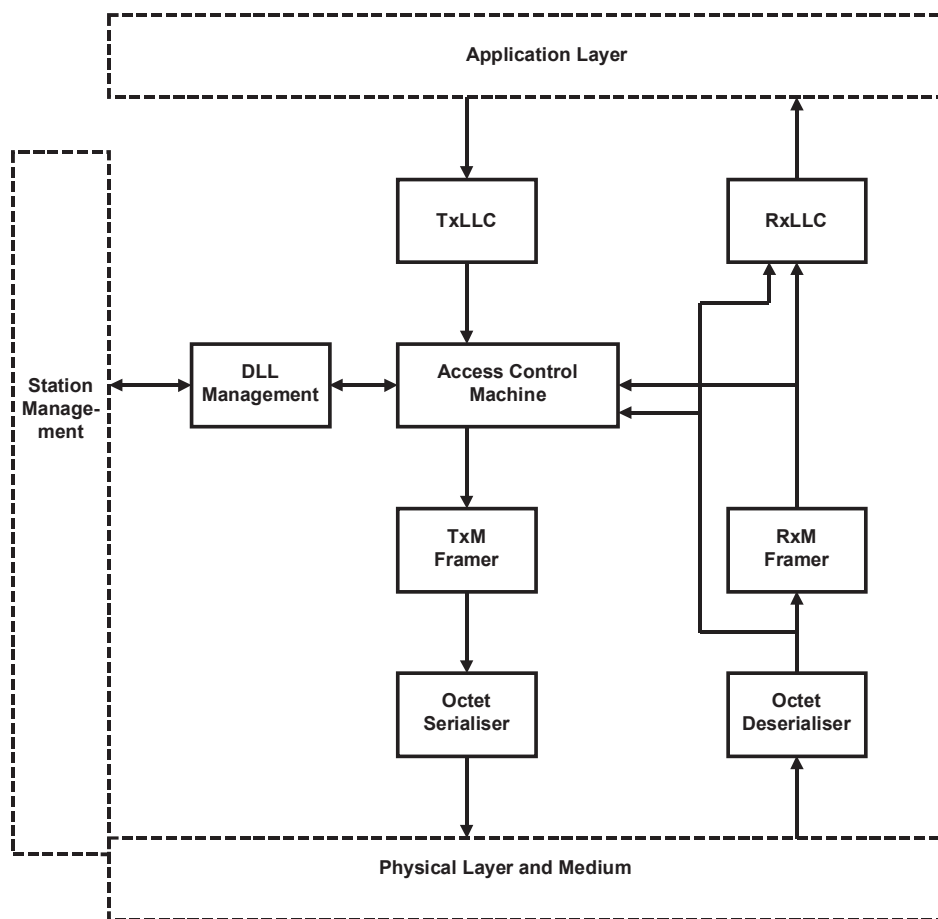


Figure 2 – Data-link layer internal architecture

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.

NOTE 1 Only DLSDUs with scheduled as their QoS parameter may be included in a scheduled DLPDU.

NOTE 2 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).

This subclause 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 0–99, values from 100–254 are reserved.

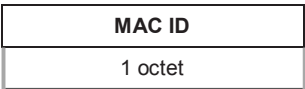


Figure 3 – Basic structure of a MAC ID address

Table 2 specifies the use of MAC ID addresses.

Table 2 – 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.

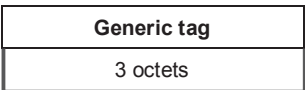


Figure 4 – Basic structure of a generic tag address

NOTE 1 Generic tags are used to send connected DLSDUs and may be associated with any of the available QoS priorities (Scheduled, High, Low).

NOTE 2 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.

Fixed tag service	Destination MAC ID
1 octet	1 octet

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 3. 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 3 – 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

The following subclauses 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 4), 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.

4.4.2 Data encoding rules

The M_Symbols transferred at the DLL to PhL interface are of four types designated 0, 1, +, -. They will be encoded inside the PhL into appropriate Ph_Symbols as shown in Table 4. The M_0 and M_1 will be encoded into Ph_Symbols that represent Manchester biphas L data encoding rules. The M_ND symbols are non-data symbols to allow for the creation of unique data patterns used for start and end delimiters. The example of signal voltage waveform as it might be applied to an electrically-conductive medium is shown in an idealized form in Figure 6 to provide an example of the data encoding rules shown in Table 4.

Table 4 – Data encoding rules

Data bits (common name)	M_Symbol representation	Ph_Symbol encoding	Manchester encoded
data “zero”	M_0 or {0}	{L,H}	0
data “one”	M_1 or {1}	{H,L}	1
“non_data+”	M_ND+ or {+}	{H,H}	no data
“non_data-”	M_ND- or {-}	{L,L}	no data

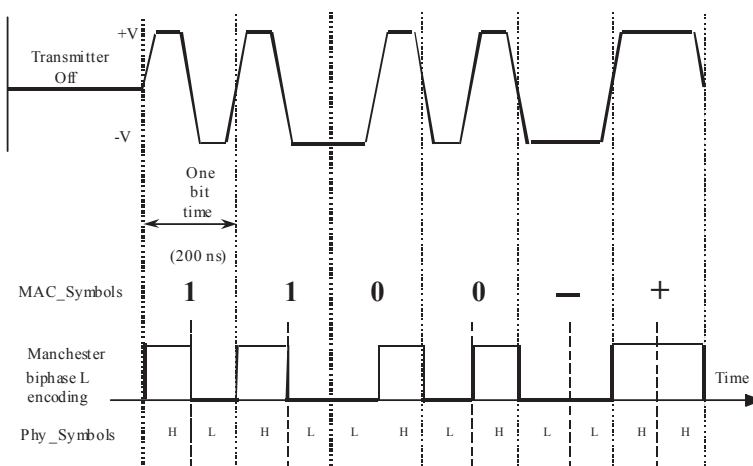


Figure 6 – M_symbols and Manchester encoding at 5 MHz

4.4.3 DLL to PhL interface

4.4.3.1 General

The DLL to PhL interface need not be exposed in the implementation of any PhL variant. This interface may be internal to the node. If conformance to the DLL to PhL interface is claimed, it shall conform to the requirements of this clause.

4.4.3.2 Ph_lock_indication

ph_lock_indication shall provide an indication of either data lock or Ph_Symbol synchronization by the PhL. Valid states for ph_lock_indication shall be true and false. ph_lock_indication shall be true whenever valid Ph_Symbols are present at the PhL interface and the PhL to DLL interface timing of M_Symbols conforms to

the PhL requirements. It shall be false between frames (when no Ph_Symbols are present on the medium) or whenever data synchronization is lost or the timing fails to conform to the PhL requirements. ph_lock_indication shall be true prior to the beginning of the start delimiter.

4.4.3.3 Ph_frame_indication

ph_frame_indication shall provide an indication of a valid data frame from the PhL. Valid states for ph_frame_indication shall be true and false. ph_frame_indication shall be true upon ph_lock_indication = true and reception of the first valid start delimiter. ph_frame_indication shall be false at reception of next M_ND symbol (following the start delimiter) or ph_lock_indication = false.

NOTE This signal provides octet synchronization to the DLL.

4.4.3.4 Ph_carrier_indication

ph_carrier_indication shall represent the presence of a signal carrier on the medium. The ph_carrier_indication shall be true if internal PhL identification of signal carrier has been true during any of the last 4 M_symbol times and it shall be false otherwise.

4.4.3.5 Ph_data_indication

ph_data_indication shall represent the M_Symbols that are decoded from the PhL internal representations such as those shown in Table 4. Valid M_symbols shall be M_0 {0}, M_1 {1}, M_ND+ {+}, and M_ND- {-} as shown in Table 5.

Table 5 – M Data symbols

Data bits (common name)	M_Symbol representation
data “zero”	M_0 or {0}
data “one”	M_1 or {1}
“non_data+”	M_ND+ or {+}
“non_data-”	M_ND- or {-}

The ph_data_indication shall represent the M_Symbols as decoded within the PhL whenever ph_lock_indication is true.

4.4.3.6 Ph_status_indication

ph_status_indication shall represent the delimiter status of the frame which was received from the PhL as shown in Table 6. Valid symbols shall be Normal, Abort, and Invalid. ph_status_indication shall indicate Normal after reception of a frame (ph_frame_indication = true) composed of a start delimiter, valid data (no M_ND symbols) and an end delimiter. ph_status_indication shall indicate Abort after reception of a frame (ph_frame_indication = true) composed of a start delimiter, valid data, and a second start delimiter. ph_status_indication shall indicate Invalid after reception of a frame (ph_frame_indication = true) composed of a start delimiter and the detection of any M_ND symbol which was not part of a start or end delimiter.

Table 6 – Truth table for ph_status_indication

Ph_status_indication	Ph_frame_indication	Start delimiters in a single frame	End delimiter detection	Any non-delimiter Manchester violations
Normal	true	1	true	false
Abort	true	2	don't care	false
Invalid	true	1	don't care	true

4.4.3.7 Ph_data_request

ph_data_request shall present the M_Symbols to be transmitted. Valid symbols shall be M_0, M_1, M_ND+ or M_ND- as shown in Table 5. ph_data_request shall indicate M_0 when no data is to be transmitted (and ph_frame_request = false).

4.4.3.8 Ph_frame_request

ph_frame_request shall be true when ph_data_request presents M_Symbols to be encoded into the appropriate Ph_Symbols by the PhL, and shall be false when no valid M_Symbols are to be transferred to the PhL.

4.5 Functional classes

Implementations of this protocol fall into two broad classes:

- connection originators (sometimes called Scanners);
- connection responders or connection targets (sometimes called Adaptors).

Devices with connection origination capability will include functions such as Scheduling object, support for external Configuration Tools and usually a Keeper object.

Connection responders are typically simpler devices which do not need the full set of capabilities.

This specification does not mandate particular groupings of functions for classes of implementation devices.

NOTE Grouping of services and functions into implementation classes may be done by separate specifications for profiles.

5 General structure and encoding of PhIDUs and DLPDUs and related elements of procedure

5.1 Overview

The DLL and its procedures are those necessary to provide the services offered to DLS user by using the services available from the PhL. The main services of this protocol are

- a) provision of timely access opportunities to support scheduled data-transfer DL-services on one link and on an extended multi-link network;
- b) the packing of multiple DLS-user transfers into a single PhL transfer unit (DLPDU) to efficiently use each transfer opportunity;
- c) the efficient distribution of DLS-user data from a publishing DLS-user to a set of subscribing DLS-users;
- d) the provision of a synchronized sense of internal time among the DLEs and available to the DLS-users of the extended link;
- e) the standardized availability of local DL-address, buffer and queue management capabilities.

5.2 Media access procedure

The primary task of the Data Link Layer is to determine, in co-operation with other Data Link Layers on the same link, the granting of permission to transmit on the medium. At its upper interface, the DLL provides services to receive and deliver service data units for the DLS user and Station Management.

The DLL protocol is based on a fixed, repetitive time cycle called a network update time (NUT). The NUT is maintained in close synchronism among all nodes on the link. A node is not permitted access to transmit on the medium if its NUT does not agree with the NUT currently being used on a link wide basis. Different links may have different NUT duration.

Each node contains its own timer synchronized to the NUT for the local link. Media access is determined by local sub-division of the NUT into access slots. Access to the medium is in sequential order based on the MAC ID of the node. Specific behaviors have been incorporated into the access protocol allowing a node

which temporarily assumes a MAC ID of zero to perform link maintenance. The MAC ID numbers of all nodes on a link are unique. Any DLL detecting the presence of a duplicate MAC ID shall immediately stop transmitting.

An implicit token passing mechanism is used to grant access to the medium. Each node monitors the source MAC ID of each DLPDU received. At the end of a DLPDU, each DLL sets an "implicit token register" to the received source MAC ID + 1. If the implicit token register is equal to the local MAC ID, then the node shall transmit one DLPDU containing one or more Lpackets with data or a null DLPDU. In all other cases, the node watches for either a new DLPDU from the node identified by the "implicit token register" or a time-out value if the identified node fails to transmit. In each case, the "implicit token" is automatically advanced to the next MAC ID. All nodes have the same value in their "implicit token register" preventing collisions on the medium.

The time-out period (called the "slot time") is based on the amount of time required for

- the current node to hear the end of the transmission from the previous node;
- the current node to begin transmitting;
- the next node to hear the beginning of the transmission from the current node.

The slot time is adjusted to compensate for the total length of the medium since the propagation delay of the medium effects the first and last item on the previous list.

NOTE The calculation of slot time is specified in 8.2.

Each NUT is divided into three major parts: scheduled, unscheduled, and guardband as shown in Figure 7. This sequence is repeated in every NUT. The implicit token passing mechanism is used to grant access to the medium during both the unscheduled and scheduled intervals.

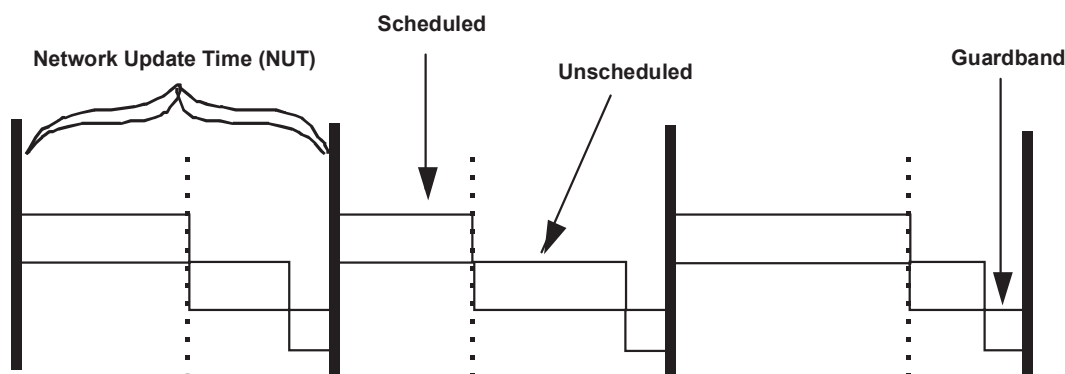


Figure 7 – NUT structure

During the scheduled part of the NUT, each node, starting with node 0 and ending with node SMAX, gets a chance to transmit time-critical (scheduled) data. SMAX is the MAC ID of the highest numbered node that has access to the medium during the scheduled part of the NUT. Every node between 0 and SMAX has only one opportunity to send one DLPDU of scheduled data in each NUT. The opportunity to access the medium during the scheduled time is the same for each node in every NUT. This allows data that is transmitted during the scheduled portion of the NUT to be sent in a predictable and deterministic manner. Figure 8 shows how the permission to transmit is granted during the scheduled time. The DLS-user regulates the amount of data that each node may transmit during this scheduled token pass.

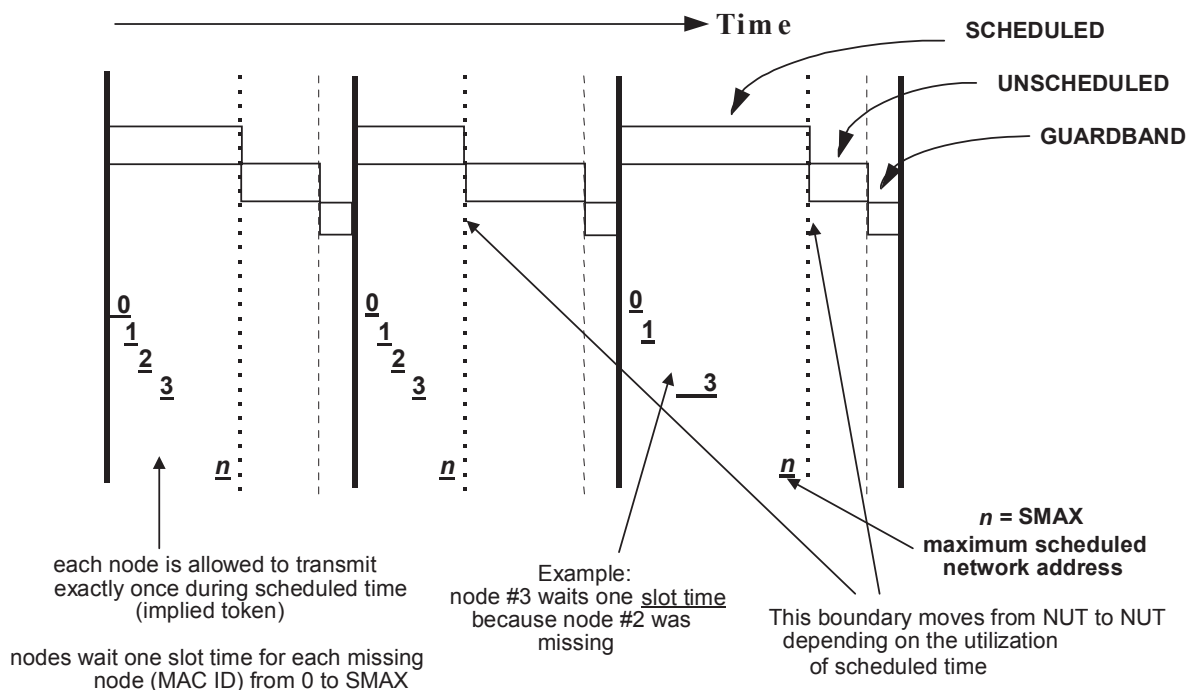


Figure 8 – Media access during scheduled time

During the unscheduled part of the NUT, each node from 0 to UMAX shares the opportunity to transmit one DLPDU of non-time critical data in a round robin fashion, until the allocated NUT duration is exhausted. UMAX is the MAC ID of the highest numbered node that has access to the medium during the unscheduled part of the NUT. The round robin method of access opportunity enables every node between 0 and UMAX to have zero, one or many opportunities to send unscheduled data depending on how much of the NUT remains after the completion of the scheduled time. Variations in scheduled traffic means the opportunity to access the medium during the unscheduled time may be different for each node in every NUT. Figure 9 shows how the permission to transmit is granted during the unscheduled time. The MAC ID of the node that goes first in the unscheduled part of the NUT is incremented by 1 for each NUT. The unscheduled token begins at the MAC ID specified in the unscheduled start register (USR) of the previous moderator DLPDU. The USR increments by one modulo (UMAX+1) each NUT. If the USR reaches UMAX before the guardband, it returns to zero and the token pass continues.