INTERNATIONAL STANDARD

ISO/IEC 9314-7

First edition
1998-08

Information technology –
Fibre distributed data interface (FDDI) –

Part 7:
Physical Layer Protocol (PHY-2)

Reference number
ISO/IEC 9314-7:1998(E)

INTERNATIONAL STANDARD

ISO/IEC 9314-7

First edition
1998-08

Information technology – Fibre distributed data interface (FDDI) – Part 7: Physical Layer Protocol (PHY-2)

PRICE CODE U

For price, see current catalogue

CONTENTS

Figures

Tables

FOREWORD

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9314-7 was prepared by Joint Technical Committee ISO/IEC JTC 1 *Information technology*, Subcommittee SC 25*, Interconnection of information technology equipment.*

ISO/IEC 9314 consists of the following parts, under the general title *Information technology – Fibre Distributed Data Interface (FDDI):*

– *Part 1: Token Ring Physical Layer Protocol (PHY) (1989)*
– *Part 2: Token Ring Media Access Control (MAC) (1989)*
– *Part 3: Physical Layer Medium Dependent (PMD) (1990)*
– *Part 4: Single Mode Fibre Physical Layer Medium Dependent (SMF-PMD)* [1)]
– *Part 5: Hybrid Ring Control (HRC) (1995)*
– *Part 6: Station Management (SMT)*
– *Part 7: Physical Layer Protocol (PHY-2)*
– *Part 8: Media Access Control-2 (MAC-2)*
– *Part 9: Low-Cost Fibre – Physical Medium Dependent (LCF-PMD) (under consideration)*
– *Part 10: Token Ring Twisted Pair Physical layer Medium Dependent (TP-PMD) (under consideration)*
– *Part 13: Conformance Test Protocol Implementation Conformance Statement Proforma (CT-PICS)*
– *Part 20: Physical Medium Dependent Conformance Testing (PMD-ATS) (under consideration)*
– *Part 21: Physical Layer Protocol Conformance Testing (PHY-ATS) (under consideration)*
– *Part 25: Abstract Test Suite for FDDI – Station Management Conformance Testing (SMT-ATS)*
– *Part 26: Media Access Control Conformance Testing (MAC-ATS) (under consideration)*

1) To be published.

INTRODUCTION

The Fibre Distributed Data Interface (FDDI), ISO/IEC 9314, is intended for use in a high-performance general purpose multi-node network and is designed for efficient operation with a peak data rate of 100 Mbit/s. It uses a Token Ring architecture with optical fibre as the transmission medium. FDDI provides for hundreds of nodes operating over an extent of tens of kilometers.

The Physical Layer Protocol (PHY) specifies the upper sublayer of the Physical Layer for the FDDI. As such, it presents the specifications and services provided for conforming FDDI attachment devices. PHY specifies the data encode and decode, framing, and clocking requirements. PHY also specifies the elasticity buffer, smoothing, and repeat filter functions.

When the set of basic FDDI standards, ISO/IEC 9314, is completed it will include the following standards:

a) A Media Access Control (MAC), which specifies the lower sublayer of the Data Link Layer of ISO/IEC 9314,

b) A Physical Layer Media Dependent (PMD), which specifies the lower sublayer of the Physical Layer of ISO/IEC 9314,

c) A Station Management (SMT), which specifies the local portion of the system management application process of ISO/IEC 9314.

A number of extensions to ISO/IEC 9314 are completed or in process. One extension, ISO/IEC 9314-5, for Hybrid Ring Control (HRC) commonly known as FDDI-II, extends the capability of FDDI to handle isochronous data streams at a multiplicity of data rates. Another extension, ISO/IEC 9314-4, provides for a single-mode optical fibre version of PMD (SMF-PMD) and will permit optical links of up 60 km.

Other extensions, addressing alternate PMDs, provide low-cost attachments for use in concentrator-to-workstation environments.

This part of ISO/IEC 9314 for PHY-2 is an enhancement to the original FDDI standard on PHY (ISO 9314-1). It is referred to as PHY-2 when it is necessary to distinguish it from the original PHY. Changes include those identified in footnotes to ISO 9314-1 as areas that the standards committee intended to change as well as changes that were required for extensions to FDDI, such as FDDI-II. PHY-2 also includes editorial corrections and clarifications.

INFORMATION TECHNOLOGY — FIBRE DISTRIBUTED DATA INTERFACE (FDDI) —

Part 7: Physical Layer Protocol (PHY-2)

1 Scope

This part of ISO/IEC 9314 specifies the Physical Layer Protocol (PHY), the upper sublayer of the Physical Layer, for Fibre Distributed Data Interface (FDDI).

FDDI provides a high-bandwidth (100 Mbit/s), general-purpose interconnection among information processing systems, subsystems and peripheral equipment, using fibre optics or other transmission media. FDDI can be configured to support a sustained data transfer rate of at least 80 Mbit/s (10 Mbyte/s). FDDI provides connectivity for many nodes distributed over distances of many kilometers in extent. Certain default parameter values for FDDI (e.g. timer settings) are calculated on the basis of up to 1 000 transmission links or up to 200 km total fibre-path length (typically corresponding to 500 nodes and 100 km of dual fibre cable, respectively); however, the FDDI protocols can support much larger networks by increasing these parameter values.

As shown in figure 1, FDDI consists of

a) Physical Layer (PL), which is divided into two sublayers:

   1) A Physical Medium Dependent (PMD), which provides the digital baseband point-to-point communication between nodes in the FDDI network. The PMD provides all services necessary to transport a suitably coded digital bit stream from node to node. The PMD defines and characterizes the fibre-optic drivers and receivers, medium-dependent code requirements, cables, connectors, power budgets, optical bypass provisions, and physical-hardware-related characteristics. It specifies the point of interconnectability for conforming FDDI attachments. The initial PMD standard defines attachment to multi-mode fibre. Alternative PMD sublayer standards are being developed for attachment to other transmission media and for mapping to Synchronous Optical Network (SONET),

   2) A Physical Layer Protocol (PHY), which provides connection between the PMD and the Data Link Layer. PHY establishes clock synchronization with the upstream code-bit data stream and decodes this incoming code-bit stream into an equivalent symbol stream for use by the higher layers. PHY provides encoding and decoding between data and control indicator symbols and code bits, medium conditioning and initializing, the synchronization of incoming and outgoing code-bit clocks, and the delineation of octet boundaries as required for the transmission of information to or from higher layers. Information to be transmitted on the medium is encoded by the PHY using a group transmission code. The definition of PHY is contained in this part of ISO/IEC 9314.

b) A Data Link Layer (DLL), which is divided into two or more sublayers:

   1) An optional Hybrid Ring Control (HRC), which provides multiplexing of packet and circuit switched data on the shared FDDI medium. HRC comprises two internal components, a Hybrid Multiplexer (H-MUX) and an Isochronous MAC (I-MAC). H-MUX maintains a synchronous 125 µs cycle structure and multiplexes the packet and circuit switched data streams, and I-MAC provides access to circuit switched channels,

   2) A Media Access Control (MAC), which provides fair and deterministic access to the medium, address recognition, and generation and verification of frame check sequences. Its primary function is the delivery of packet data, including frame generation, repetition, and removal,

   3) An optional Logical Link Control (LLC), which provides a common protocol for any required packet data adaptation services between MAC and the Network Layer. LLC is not specified by FDDI,

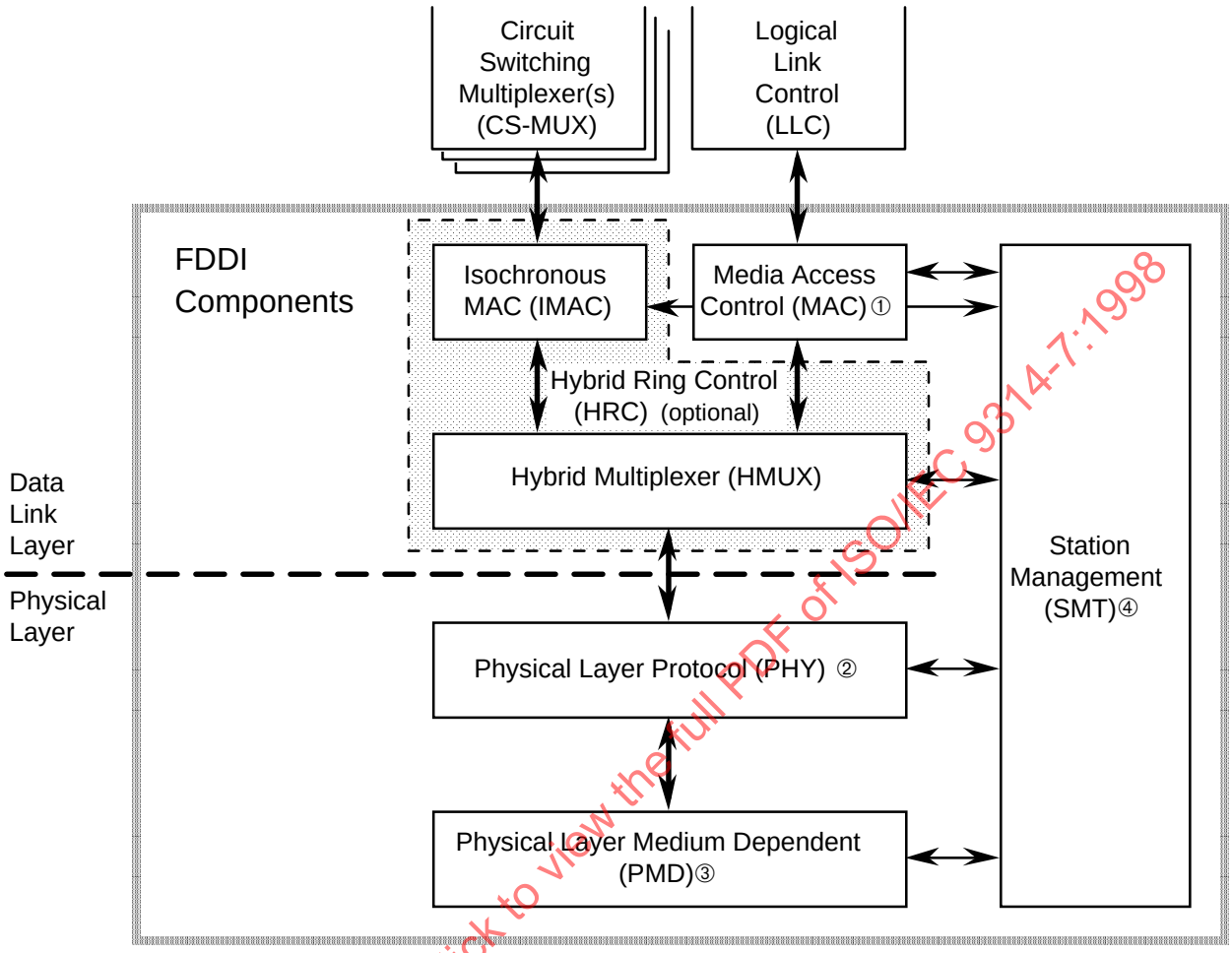
4) An optional Circuit Switching Multiplexer (CS-MUX), which provides a common protocol for any required circuit data adaptation services between I-MAC and the Network Layer. CS-MUX is not specified by FDDI.

c) A Station Management (SMT), which provides the coordination necessary at the node level to manage the processes under way in the various FDDI layers such that a node may work cooperatively on a ring. SMT provides services such as control of configuration management, fault isolation and recovery, and scheduling policies.

The definition of PHY as contained in this part of ISO/IEC 9314 is designed to be as independent as possible from the actual physical medium.

This part of ISO/IEC 9314 is an optional alternative to the original part of ISO/IEC 9314 on PHY (ISO 9314-1) for implementations without an (optional) HRC, and is required for implementations with an HRC. Implementations that conform to this part of ISO/IEC 9314 shall also be interoperable with implementations that conform to ISO 9314-1 if the additional capability of Hybrid mode operation (as defined in this part of ISO/IEC 9314) is not being used. Implementers are encouraged to read ISO 9314-1 in addition to this part of ISO/IEC 9314.

The set of FDDI standards specifies the interfaces, functions and operations necessary to ensure interoperability between conforming FDDI implementations. This part of ISO/IEC 9314 is a functional description. Conforming implementations may employ any design technique that is interoperable.



① MAC-2 with HRC; MAC or MAC-2 otherwise.

② PHY-2 with HRC; PHY or PHY-2 otherwise.

③ PMD, SMF-PMD, TP-PMD or LCF-PMD.

④ SMT-2 with HRC; SMT or SMT-2 otherwise.

Figure 1 – Structure of FDDI standards

2 Normative references

The following standards contain provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9314. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this part of ISO/IEC 9314 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO 9314-1: 1989, *Information processing systems – Fibre Distributed Data Interface (FDDI) – Part 1: Token Ring Physical Layer Protocol (PHY)*

ISO 9314-2: 1989, *Information processing systems – Fibre Distributed Data Interface (FDDI) – Part 2: Token Ring Media Access Control (MAC)*

ISO/IEC 9314-3: 1990, *Information processing systems – Fibre Distributed Data Interface (FDDI) – Part 3: Physical Layer Medium Dependent (PMD)*

ISO/IEC 9314-4, *Information technology – Fibre Distributed Data Interface (FDDI) – Part 4: Single Mode Fibre Physical Layer Medium Dependent (SMF-PMD)* [1)]

ISO/IEC 9314-5:1995, *Information technology – Fibre Distributed Data Interface (FDDI) – Part 5: Hybrid Ring Control (HRC)*

ISO/IEC 9314-6, *Information technology – Fibre Distributed Data Interface (FDDI) – Part 6: Station Management (SMT)*

ISO/IEC 9314-8, *Information technology – Fibre Distributed Data Interface (FDDI) – Part 8: Media Access Control (MAC-2)*

ISO/IEC 9314-9:199X, *Fibre Distributed Data Interface (FDDI) – Part 9: Token ring low-cost fibre physical layer medium dependent (LCF-PMD)*

ISO/IEC 9314-10:199X, *Fibre Distributed Data Interface (FDDI) – Part 10: Token ring twisted pair physical layer medium dependent (TP-PMD)*

3 Definitions

For the purposes of this part of ISO/IEC 9314, the following definitions apply. In some cases these definitions may duplicate those contained in other parts of ISO/IEC 9314. Such definitions are included for completeness and to improve readability. In certain cases, definitions herein may slightly update those contained in the earlier published parts of ISO/IEC 9314 to improve their clarity.

3.1 basic mode: The mode of ring operation where MAC PDUs (frames and tokens) are directly transmitted by PHY.

3.2 byte: A pair of symbols on an even-symbol boundary.

3.3 code bit: The smallest signalling element used by the Physical Layer for transmission on the medium.

3.4 code group: A Protocol Data Unit transmitted between cooperating PHY entities on a Physical Link, consisting of a specific sequence of five code bits representing a symbol.

3.5 concentrator: A node which provides connections for multiple subordinate nodes in an FDDI network. A concentrator has two or more Physical Layer attachments and may or may not have one or more Data Link Layer entities.

3.6 Connection Management (CMT): That portion of the Station Management (SMT) function that controls network insertion, removal, and connection of PHY and Data Link Layer entities (MAC or HRC) within a node.

3.7 counter-rotating: An arrangement whereby two signal paths operate in opposite directions in a ring topology.

---

1) To be published.

3.8 cycle: A Protocol Data Unit transmitted between cooperating HRC entities on a ring, consisting of a fixed number of symbols in each 125 µs interval.

3.9 entity: An active service or management element within an Open System Interconnection (OSI) layer, or sublayer.

3.10 fibre optics: A technology whereby signals are transmitted over an optical waveguide medium through the use of light-generating transmitters and light-detecting receivers.

3.11 frame: A Protocol Data Unit transmitted between cooperating MAC entities on a ring, consisting of a variable number of octets and control symbols.

3.12 Hybrid mode: The mode of ring operation where HRC PDUs (cycles) are transmitted by PHY.

3.13 Hybrid Ring Control (HRC): The Data Link Layer entity responsible for multiplexing of packet and circuit switched data, and providing access to circuit switched channels, in an FDDI logical ring.

3.14 logical ring: The set of FDDI Data Link Layer entities (HRC and MAC) serially connected to form a single ring. The FDDI network topology can form two counter-rotating logical rings, however, some subsets of this topology can only form a single logical ring.

3.15 Media Access Control (MAC): The Data Link Layer entity responsible for scheduling and routing packet data transmissions in an FDDI logical ring.

3.16 network (FDDI network): A collection of FDDI nodes interconnected to form a trunk, or a tree, or a trunk with multiple trees. This topology is sometimes called a dual ring of trees.

3.17 node: A collection of Physical Layer (e.g. PMD and PHY) and optional Data Link Layer (e.g. MAC and HRC) entities within an FDDI network, capable of repeating information and optionally of transmitting and receiving information, and managed by one SMT entity.

3.18 non-return to zero (NRZ): A coding technique where one polarity level represents a logical "1" (one) and the opposite polarity level represents a logical "0" (zero).

3.19 non-return to zero, invert on ones (NRZI): A coding technique where a polarity transition represents a logical "1" (one) and the absence of a polarity transition denotes a logical "0" (zero).

3.20 octet: A data unit composed of eight ordered binary bits. An octet is represented in FDDI as a pair of data symbols.

3.21 Physical Connection: The full-duplex physical layer association between adjacent PHY entities (in concentrators, repeaters, or stations) in an FDDI ring, i.e. a pair of Physical Links.

3.22 Physical Layer Medium Dependent (PMD): The Physical Layer entity responsible for delivering a code bit stream produced by a PHY entity to the physically adjacent PHY entity, attached via fibre optics or other media, in an FDDI network.

3.23 Physical Layer Protocol (PHY): The Physical Layer entity responsible for delivering a symbol stream produced by an upstream DLL entity (MAC or HRC) to the logically adjacent downstream DLL entity in an FDDI network.

3.24 Physical Link: The simplex path (via PMD and attached medium) from the transmit function of one PHY entity to the receive function of an adjacent PHY entity (in concentrators, repeaters, or stations) in an FDDI ring.

3.25 Port: A PHY entity and a PMD entity in a node, together creating a PHY/PMD pair, that may connect to the optical fibre or other media and provide one end of a physical connection with another node.

3.26 primitive: An element of the services provided by one entity to another.

3.27 Protocol Data Unit (PDU): The unit of information transfer between communicating peer layer entities. It may contain control information, address information, data (e.g. an SDU from a higher layer entity), or any combination of the three. The FDDI PHY PDUs are code groups.

3.28 quartet: A data unit composed of four ordered binary bits. A quartet is represented in FDDI as a single data symbol.

3.29 receive: The action of a node that consists of accepting an information stream (e.g. frame, token, cycle or control sequence) from the medium. The node receiving the information stream may examine it and selectively copy it as appropriate.

3.30 repeat: The action of a node that consists of receiving an information stream from an upstream node and reproducing it on the medium to a downstream node. The node repeating the information stream may examine it and selectively copy or modify it as appropriate.

3.31 ring: A closed loop consisting of one or more stations connected by a physical medium wherein information is passed sequentially between active stations, each station in turn examining or copying and repeating the information, finally returning it to the originating station.

3.32 Service Data Unit (SDU): The unit of data transfer between a service user and a service provider.

3.33 services: A set of functions provided by one OSI layer or sublayer entity, for use by a higher layer or sublayer entity or by management entities. Data services are provided to a higher layer or sublayer entity; management services are provided to a management entity.

3.34 station: An addressable logical and physical node in an FDDI network, capable of transmitting, repeating and receiving information. An FDDI station has one or more PHY and PMD entities, zero or more HRC entities, one or more MAC entities, and one SMT entity.

3.35 Station Management (SMT): The supervisory entity within an FDDI node that monitors and controls the other FDDI entities in the node.

3.36 symbol: The smallest signalling element used by the Data Link Layer (DLL). The symbol set consists of 16 data symbols and 9 control symbols. Each symbol corresponds to a specific code group to be transmitted by PHY.

3.37 transmit: The action of a node that consists of generating an information stream (e.g. frame, token, cycle or control sequence) and placing it on the medium.

3.38 Unit Interval (UI): The transmission time for a fixed length signalling element (e.g. a code bit or a symbol).

4 Conventions and abbreviations

4.1 Conventions

The terms SMT, MAC, HRC, PHY and PMD, when used without modifiers, refer specifically to the local FDDI entities within a node. When this edition is to be distinguished from the first edition (ISO 9314-1: 1989) it shall be referred to as PHY-2; the term PHY is used generically where either version may be applicable.

Low lines (e.g. control_action) are used as a convenience to mark the name of signals, functions, etc., which might otherwise be misinterpreted as independent individual words if they were to appear in text.

The use of a period (e.g. PH_UNITDATA.request) is equivalent to the use of a low line except that a period is used as an aid to distinguish modifier words appended to an antecedent expression.

Subscripts or other object selectors are denoted by square brackets in text (e.g. aggregate object[subscript]).

An overbar (¯) after a decimal fraction denotes a continued fraction (e.g. $0{,}04\overline{\ }$ = 0,044 444 …).

A vertical stroke (|) in a logical expression denotes a logical ‘OR’.

An ampersand (&) in a logical expression denotes a logical ‘AND’.

A tilde (~) in a logical expression denotes a logical ‘NOT’.

Optional capabilities are distinguished from required capabilities by the use of dashed lines in drawings or curved braces in text (e.g. required capability { | optional capability }).

Comments in state machine footnotes are denoted by double angle brackets (e.g. condition «comment»).

4.2 Abbreviations

PMD	Physical Layer Medium Dependent (see ISO/IEC 9314-3, 9314-4, 9314-9 or 9314-10)
PHY	Physical Layer Protocol (see ISO/IEC 9314-7 or this standard)
HRC	Hybrid Ring Control (see ISO/IEC 9314-5)
MAC	Media Access Control (see ISO/IEC 9314-2 or 9314-8)
SMT	Station Management (see ISO/IEC 9314-6)
CMT	Connection Management
SONET	Synchronous Optical Network
DLL	Data Link Layer
ALS	Active Line State
CLS	Cycle Line State
HLS	Halt Line State
ILS	Idle Line State
MLS	Master Line State
NLS	Noise Line State
QLS	Quiet Line State
LSU	Line State Unknown
NRZ	Non Return to Zero
NRZI	Non Return to Zero, Invert on Ones
PI	Primary In
PO	Primary Out
SI	Secondary In
SO	Secondary Out
UI	Unit Interval
AT_max	Maximum PHY acquisition time
LS_Max	Maximum line state change interval
Limit_ct	Current Limit Smoother extension (in symbols)
Target_ct	Current Target Smoother extension (in symbols)
Out_ct	Number of output symbols
C_Flag	Indicates Hybrid mode operation
S_Flag	Indicates synchronization established in Hybrid mode
T_Flag	Indicates current frame cannot be stripped
D_Max	Maximum ring latency
Limit_max	Maximum Limit Smoother capacity (in symbols)
Limit_cntr	One half of maximum Limit Smoother capacity (in symbols)
Target_max	Maximum Target Smoother capacity (in symbols)
Target_cntr	One half of maximum Target Smoother capacity (in symbols)
Target_th	Current Target Smoother threshold (in symbols)
PA_max	Maximum Hybrid mode preamble length (in symbols) before loss of synchronization
MIC	Media Interface Connector
P_max	Maximum number of Ports in a logical ring
SD_max	Maximum effective starting delimiter delay contribution of this attachment
SD_min	Minimum starting delimiter delay through this attachment
ms	millisecond
µs	microsecond
ns	nanosecond

5 General description

An FDDI network consists of a set of nodes (e.g. stations) connected by optical transmission media into one or more logical rings. A logical ring consists of a set of stations connected as an alternating series of nodes and transmission media to form a closed loop. Information is transmitted sequentially, as a stream of suitably encoded symbols, from one active node to the next. Each node generally regenerates and repeats each symbol and serves as the means for attaching one or more devices to the ring for the purpose of communicating with other devices on the ring.

Two kinds of data service can be provided in a logical ring: packet service and circuit service. For packet service, a given station (the one that has access to the medium) transmits information onto the ring as a series of data packets, where each packet circulates from one station to the next. The addressed destination station(s) copies the packets as they pass. Finally, the station that transmitted the packets effectively removes them from the ring. For circuit service, some of the logical ring bandwidth is allocated to independent channels. Two or more stations can simultaneously communicate via each channel. The structure of the information stream within each channel is determined by the stations sharing that channel.

The basic building block of an FDDI network is a Physical Connection as shown in figure 2. A Physical Connection connects a pair of Physical Layer endpoints (PHYs) in FDDI nodes through a transmission medium. A peer Physical Connection connects adjacent nodes in the FDDI trunk ring. A hierarchical Physical Connection connects a master PHY in a concentrator to a subordinate PHY in another node, to create a branching topology. Physical Connections may be interconnected within nodes, via attached Data Link Layer entities (MAC or HRC)s or other means, to create the FDDI network.

A Physical Connection consists of a pair of Physical Links carrying signals in opposite directions between the pair of PHYs. A peer Physical Connection in the FDDI trunk ring consists of a Primary Link and a Secondary Link. A Primary Link consists of the output end, called Primary Out (PO), of a PHY, communicating over a Primary medium to the input end, called Primary In (PI), of a second PHY. A Secondary Link consists of the output end, called Secondary Out (SO), of the second PHY communicating over a Secondary medium to the input end, called Secondary In (SI), of the first PHY.

An FDDI network consists of a theoretically unlimited number of connected nodes. SMT establishes the Physical Connections between nodes, and the correct internal node configurations, to create an FDDI network of logical rings. The method of actual physical attachment of nodes to the FDDI network will vary and is dependent on specific application requirements. The function of each node is not specified in this part of ISO/IEC 9314, but is determined by its specific application and management requirements. Application requirements are not specified by any part of ISO/IEC 9314; however, management requirements are specified in ISO/IEC 9314-6.

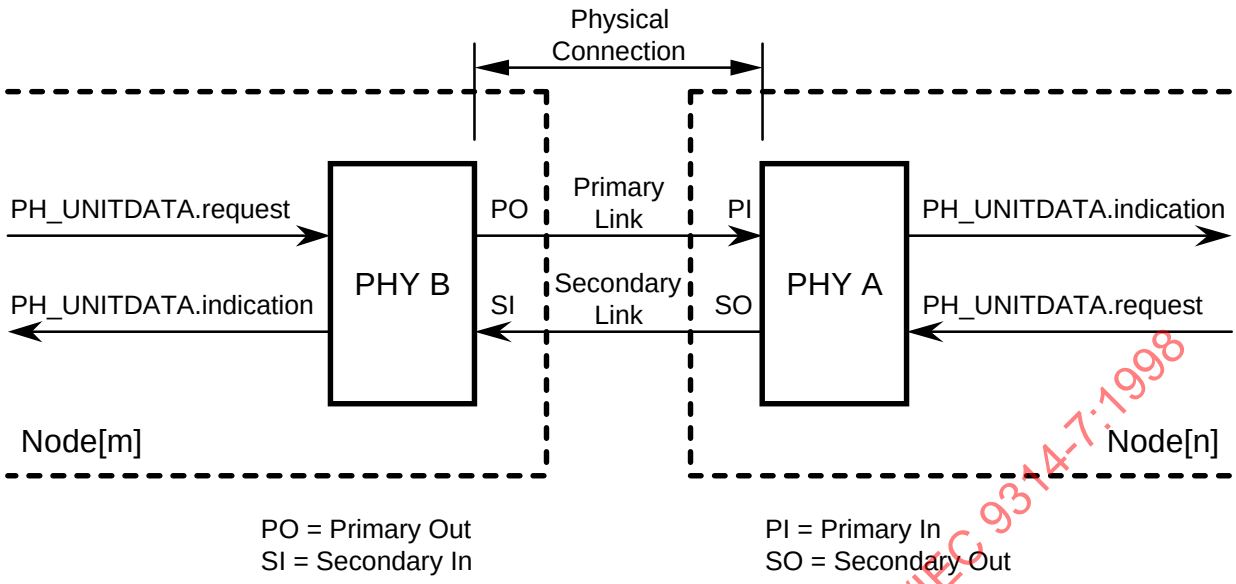


Figure 2 – Peer Physical Connection example

6 Services

This clause specifies the services provided by PHY and the services required by PHY. The following services are defined:

a) Data services provided by PHY to the local Data Link Layer (DLL) entity (MAC or HRC), denoted by a PH_ prefix

b) Data services required by PHY from the local PMD entity, denoted by a PM_ prefix

c) Management services provided by PHY to the local SMT entity, denoted by a SM_PH_ prefix

All services defined in this clause are mandatory.

The definition of these services is for reference purposes only, and is not intended to specify an implementation of the PHY interfaces. Any implementation technique that causes the same external behaviour of the protocol is equally valid.

6.1 PHY-to-DLL services

This subclause specifies the services provided at the interface between the PHY and the Data Link Layer (DLL) entities, to support the exchange of PDUs among peer DLL entities (MAC or HRC). Additional detail is provided in the parts of ISO/IEC 9314 on MAC and HRC concerning conditions that generate these primitives and DLL actions upon receipt of PHY-generated primitives.

The following primitives are defined:

PH_UNITDATA.request
PH_UNITDATA.indication
PH_INVALID.indication

All primitives defined in this subclause are mandatory.

The description of each primitive includes a description of the information that shall be passed between the PHY and Data Link Layer entities.

These services shall be 'synchronous', so that each PH_UNITDATA.indication causes exactly one PH_UNITDATA.request. Depending upon the current internal configuration of the node, the PH_UNITDATA.request may be returned to the same PHY or to a different PHY. Although these services are primarily intended as a PHY-to-DLL interface, they also serve as a PHY-to-PHY interface when repeating on a logical ring with no intervening MAC or HRC. In this case the function of the Repeat Filter is required somewhere in the repeat path within the Physical Layer (see 8.9).

To ensure correct operation of the Elasticity Buffer function, the maximum DLL PDU length shall not exceed 9 000 symbols, including the Starting Delimiter through the fourth preamble (Idle) symbol after the PDU (see 8.6).

6.1.1 PH_UNITDATA.request

This primitive transfers the symbol data stream to PHY from MAC or HRC.

6.1.1.1 Semantics of the primitive

PH_UNITDATA.request (PH_Request(symbol))

The symbol specified by PH_Request(symbol) shall be one of the following:

H, I, J, K, n, R, S, T and optionally L, Q or V, from the set of symbols defined in table 1.

PH_Request(L) is not required in Basic mode. PH_Request(H, Q or V) is not required by MAC or HRC.

NOTE 1 – When repeating in the Physical Layer, each PH_Indication(symbol) (see 6.1.2.1) is presented as a PH_Request(symbol). In this case, PH_Request(H) is required and PH_Request(Q or V) is possible.

6.1.1.2 When generated

MAC or HRC shall generate one PH_UNITDATA.request for each PH_UNITDATA.indication received from PHY.

6.1.1.3 Effect of receipt

Upon receipt of this primitive during Line_State_action(Transmit_PDR), PHY shall encode the PH_Request(symbol) into the appropriate PM_Request(NRZI code) stream.

NOTE 2 – The operation of the Smoothing and Repeat Filter functions may alter the PH_Request(symbol) stream within PHY.

6.1.2 PH_UNITDATA.indication

This primitive transfers the symbol data stream from PHY to MAC or HRC.

6.1.2.1 Semantics of the primitive

PH_UNITDATA.indication (PH_Indication(symbol))

The symbol specified by PH_Indication(symbol) shall be one of the following:

H, I, J, K, n, R, S, T and optionally L, Q or V, from the set of symbols defined in table 1.

PH_Indication(L) is not required in Basic mode, and in this case it may be indicated as PH_Indication(H or V). PH_Indication(Q) may optionally be indicated as PH_Indication(H or V). PH_Indication(V) may optionally be indicated as PH_Indication(H). When repeating in the Physical Layer, PH_Indication(H, Q or V) shall be generated when PH_Invalid is asserted.

6.1.2.2 When generated

PHY shall generate a PH_UNITDATA.indication once every symbol period, derived from the decoded PM_Indication(NRZI code) stream.

NOTE 3 – The operation of the Elasticity Buffer, Line State Detection and Smoothing functions may alter the PH_Indication(symbol) stream within PHY.

6.1.2.3 Effect of receipt

Upon receipt of this primitive, MAC or HRC shall accept a symbol from PHY, process it, and generate a corresponding PH_UNITDATA.request to PHY, conveying the resulting output symbol.

6.1.3 PH_INVALID.indication

This primitive indicates to MAC or HRC that continuity of the received symbol stream has been compromised.

6.1.3.1 Semantics of the primitive

PH_INVALID.indication (PH_Invalid)

The PH_Invalid parameter shall indicate that the PH_UNITDATA.indication symbol stream is invalid.

6.1.3.2 When generated

PHY shall generate this primitive whenever it detects a Quiet, Halt, Master or Noise Line State. In addition, PHY shall generate this primitive for other interruptions of the received symbol stream detected by PHY, including Elasticity Buffer errors, receipt of Signal_Detect(off), detection of Clock_Detect(off) (if implemented) and any other detected condition that causes potential loss or duplication (as opposed to alteration) of one or more received symbols within a MAC or HRC PDU. These additional conditions may optionally be reported as PH_Indication(H, Q or V) in Basic mode.

6.1.3.3 Effect of receipt

The effect of receipt of this primitive by MAC or HRC is not specified.

6.2 PHY-to-PMD services

This subclause specifies the services provided at the interface between the PHY and the PMD entities of the Physical layer, to support the exchange of NRZI code-bit streams between peer PHY entities. Additional detail is provided in the applicable FDDI standard on PMD (ISO/IEC 9314-3, 9314-4, 9314-9, or 9314-10) concerning conditions that generate these primitives and PMD actions upon receipt of PHY-generated primitives.

The following primitives are defined:

PM_UNITDATA.request
PM_UNITDATA.indication
PM_SIGNAL.indication

All primitives defined in this subclause are mandatory.

The description of each primitive includes a description of the information that is passed between the PHY and PMD entities.

The implementation of the PHY to PMD interface is not specified. However, an exemplary implementation of this interface is provided as an annex to ISO/IEC 9314-3.

6.2.1 PM_UNITDATA.request

This primitive transfers the NRZI data stream from PHY to PMD.

6.2.1.1 Semantics of the primitive

PM_UNITDATA.request (
PM_Request(NRZI code)
)

The data conveyed by PM_Request shall be a continuous NRZI data stream (i.e. each polarity change in PM_Request signifies a NRZI code ‘one’).

6.2.1.2 When generated

PHY continuously sends PMD the current NRZI code polarity.

6.2.1.3 Effect of receipt

The effect of receipt of this primitive by PMD is not specified.

6.2.2 PM_UNITDATA.indication

This primitive transfers the NRZI data stream to PHY from PMD.

6.2.2.1 Semantics of the primitive

PM_UNITDATA.indication (
PM_Indication(NRZI code)
)

The data conveyed by PM_Indication shall be a continuous NRZI data stream (i.e. each polarity change in PM_Indication signifies a NRZI code ‘one’).

6.2.2.2 When generated

PMD continuously sends PHY the current NRZI code polarity.

6.2.2.3 Effect of receipt

In normal (non-Loopback) mode, PM_Indication is continuously sampled by the Receive function of PHY.

6.2.3 PM_SIGNAL.indication

This primitive indicates to PHY the status of the optical signal level being received by PMD.

6.2.3.1 Semantics of the primitive

PM_SIGNAL.indication (
Signal_Detect(status)
)

The Signal_Detect(status) parameter shall indicate whether the received signal level from the medium is above (Signal_Detect(on)) or below (Signal_Detect(off)) the signal detection threshold of the receiver in PMD.

6.2.3.2 When generated

PMD continuously generates this primitive to indicate the status of the received signal.

6.2.3.3 Effect of receipt

The effect of receipt of this primitive is to assert Quiet Line State during Signal_Detect(off), and to enable detection of other line states during Signal_Detect(on).

6.3 PHY-to-SMT services

This subclause specifies the services provided at the interface between the PHY and the Station Management (SMT) entities. The services supplied by the PHY allow the local SMT entity to monitor and control the operation of PHY. The PHY-to-SMT services shall have precedence over the PHY-to-DLL services. Additional detail is provided in SMT concerning conditions that generate these primitives and SMT actions upon receipt of PHY-generated primitives.

The following primitives are defined:

SM_PH_LINE_STATE.request
SM_PH_STATUS.indication
SM_PH_CONTROL.request

All primitives defined in this subclause are mandatory.

The description of each primitive includes a description of the information that is passed between the PHY and SMT entities.

6.3.1 SM_PH_LINE_STATE.request

This primitive is generated by SMT to control the output data stream sent by PHY.

6.3.1.1 Semantics of the primitive

SM_PH_LINE_STATE.request	(Line_State_action)

The Line_State_action parameter shall be one of the following:

Transmit_Quiet. When this action is requested, PHY shall send a continuous stream of Quiet symbols to PMD. In this condition the Transmit function generates no transitions. Transmit_Quiet shall also be the default condition of the PHY Transmit function initially, or after a PHY_Reset.

WARNING – The constant PM_Request(NRZI code) polarity is not specified in this part of ISO/IEC 9314, although it may optionally be controlled in an implementation. To ensure the proper effect on the signal transmitted by PMD, SMT should also issue an appropriate SM_PM_CONTROL.request to PMD (e.g. **SM_PM_CONTROL.request(Transmit_Disable) for fibre optic media).**

Transmit_Halt. When this action is requested, PHY shall send a continuous stream of Halt symbols to PMD.

Transmit_Master. When this action is requested, PHY shall send a continuous stream of alternating Halt and Quiet symbol pairs to PMD.

Transmit_Idle. When this action is requested, PHY shall send a continuous stream of Idle symbols to PMD.

Transmit_PDR. When this action is requested, PHY shall transfer the PH_Request(symbol) stream to PMD.

NOTE 4 – The operation of the Smoothing and Repeat Filter functions can alter the PH_Request(symbol) stream within PHY.

6.3.1.2 When Generated

This primitive is generated by SMT as part of Connection Management (CMT) signalling sequences.

6.3.1.3 Effect of receipt

PHY shall send a continuous stream of the commanded symbol(s) to PMD. These primitives shall take precedence over the PHY-to-DLL primitives.

6.3.2 SM_PH_STATUS.indication

This primitive is generated by PHY to inform SMT of the status of the data stream being received from PMD. The specific items reported are defined in the following subclause.

6.3.2.1 Semantics of the primitive

SM_PH_STATUS.indication		(
		status_report
		)

The status_report parameter shall be one of the following (see 7.3 for the definition of the line states):

Quiet_Line_State_received. This parameter shall be asserted by PHY when Quiet Line State (QLS) is entered.

Halt_Line_State_received. This parameter shall be asserted by PHY when Halt Line State (HLS) is entered.

Master_Line_State_received. This parameter shall be asserted by PHY when Master Line State (MLS) is entered.

Idle_Line_State_received. This parameter shall be asserted by PHY when Idle Line State (ILS) is entered.

Active_Line_State_received. This parameter shall be asserted by PHY when Active Line State (ALS) is entered.

Cycle_Line_State_received. This parameter shall be asserted by PHY when Cycle Line State (CLS) is entered. This line state is optional unless support for Hybrid mode is implemented.

Noise_Line_State_received. This parameter shall be asserted by PHY when Noise Line State (NLS) is entered.

Line_State_Unknown. This parameter shall be asserted by PHY when any of the defined line states are exited and the entry conditions for a new line state have not yet been satisfied. This status_report parameter shall include an indication of the most recently known line state.

Link_Error_detected. This parameter shall be asserted by PHY when:

a) The current known line state is Active Line State or Cycle Line State and a V symbol is decoded. This includes receipt of any invalid code point (see table 1) or a J or K which is not decoded as part of a Starting Delimiter (see 8.5).

b) The current known line state is Idle Line State and any symbol except Q, H, I, J or K is decoded. This includes receipt of any invalid code point (see table 1) or a J or K which is not decoded as part of a Starting Delimiter (see 8.5).

NOTE 5 – ISO/IEC 9314-6 on SMT defines a set of LEM error events that is similar but not identical to the set of Link_Error_detected events defined here. The set of events defined here is intended to provide comparable error sensitivity, while avoiding spurious error indications during line state signalling and mode switching.

EB_Error_detected. This parameter shall be asserted by PHY when an Elasticity Buffer error is detected, i.e. whenever it is detected that the EB has not accurately propagated a DLL PDU. The actual EB errors than can be detected are implementer defined.

6.3.2.2 When Generated

This primitive shall be generated by PHY when the indicated condition occurs.

6.3.2.3 Effect of receipt

The effect of receipt of this primitive by SMT is not specified.

6.3.3 SM_PH_CONTROL.request

This primitive has local significance and is used by SMT to control the operation of PHY.

6.3.3.1 Semantics of the primitive

SM_PH_CONTROL.request	(Control_Action, Requested_Status)

The Control_Action parameter shall include the following: PHY_Reset, Enable_Hybrid_mode, Disable_Hybrid_mode, Begin_Loopback_mode, Cancel_Loopback_mode and Present_Status.

The Requested_Status parameter shall indicate the status_report parameter(s) to be returned to SMT in response to a Present_Status control action.

6.3.3.2 When generated

This primitive is generated by SMT to cause PHY to take the action specified by the Control_Action parameter.

6.3.3.3 Effect of receipt

The value of the Control_Action parameter shall determine the effect upon PHY as follows:

a) If the Control_Action is PHY_Reset, then PHY shall at a minimum:
   1) reset the Receive function to Clock_Detect(off) (if implemented),
   2) reset the Elasticity Buffer function,
   3) reset the Line State Detection function to Line State Unknown. If Signal_Detect(off) is asserted, then the Line State Detection function may optionally be reset to Quiet Line State,
   4) reset the Line State Detection counters,
   5) reset the Smoothing function,
   6) reset the Repeat Filter function,
   7) reset the Transmit function to Transmit_Quiet,
   8) reset Hybrid mode,
   9) report PH_Invalid to the DLL interface;

b) If the Control_Action is Enable_Hybrid_mode, then PHY shall enable the recognition and processing of cycles (HRC PDUs). This Control_Action is optional unless support for Hybrid mode is implemented;

c) If the Control_Action is Disable_Hybrid_mode, then PHY shall disable the recognition and processing of cycles (HRC PDUs). This Control_Action is optional unless support for Hybrid mode is implemented;

d) If the Control_Action is Begin_Loopback_mode, then PHY shall enter Loopback mode. The intent of this mode is to loop back the output of the Transmit function to the input of the Receive function within the PHY entity, at a point as close as possible to the interface with PMD, to permit local node testing. In this mode, PHY shall return symbols presented at the PH_UNITDATA.request interface on the PH_UNITDATA.indication interface. These symbols may be altered by the actions of the Elasticity Buffer and the Repeat Filter. PHY shall present continuous NRZI code-bit zeros to the PM_UNITDATA.request interface while in Loopback mode;

WARNING – The constant PM_Request(NRZI code) polarity is not specified in this part of ISO/IEC 9314, although it may optionally be controlled in an implementation. To ensure the proper effect on the signal transmitted by PMD, SMT should also issue an appropriate SM_PM_CONTROL.request to PMD (e.g. SM_PM_CONTROL.request (Transmit_Disable) for fibre optic media).

e) If the Control_Action is Cancel_Loopback_mode, then PHY shall leave Loopback mode;

f) If the Control_Action is Present_Status, then PHY shall generate a SM_PH_STATUS.indication to SMT that includes the status indicated by the Requested_Status parameter.

7 Facilities

7.1 Coding

7.1.1 Code bit

Peer PHY entities communicate via fixed-length code bits. A code bit is the smallest signalling element used by PHY. In the NRZI code, a code bit is represented as a transition (one), or absence of a transition (zero), in the polarity of the signal on the medium.

7.1.2 Code group

A code group is a consecutive sequence of five code bits. It is used to represent a symbol on the medium. Implicit in the definition of code group is the establishment of code group boundaries. The process of establishing a code group boundary is known as 'framing', and the established boundary is known as the 'framing boundary'.

Table 1 defines the mapping of symbols to code groups.

Table 1 – Symbol coding

Code Group		Symbol	
Decimal	**Binary**	**Name**	**Assignment**
Line State Symbols			
00	00000	Q	Quiet
04	00100	H	Halt
31	11111	I	Idle
Starting Delimiter			
24	11000	J	First symbol of JK pair
17	10001	K	Second symbol of JK pair
Embedded Delimiter			
05	00101	L	Second symbol of IL pair
Data Quartets			**Hexadecimal** **Binary**
30	11110	0	0 0000
09	01001	1	1 0001
20	10100	2	2 0010
21	10101	3	3 0011
10	01010	4	4 0100
11	01011	5	5 0101
14	01110	6	6 0110
15	01111	7	7 0111
18	10010	8	8 1000
19	10011	9	9 1001
22	10110	A	A 1010
23	10111	B	B 1011
26	11010	C	C 1100
27	11011	D	D 1101
28	11100	E	E 1110
29	11101	F	F 1111
Ending Delimiter			
13	01101	T	Terminate
Control Indicators			
07	00111	R	Reset (logical Zero or Off)
25	11001	S	Set (logical One or On)
Invalid Code Points			
01	00001	V or H	These code points shall not be transmitted because they can generate patterns that violate run length or duty cycle requirements. Streams of code points 01, 02, 08 and 16 shall be interpreted as Halt by the Line State Detection function.
02	00010	V or H	
03	00011	V	
06	00110	V	
08	01000	V or H	
12	01100	V	
16	10000	V or H	

(12345) = sequential order of code bit transmission.

7.2 Symbol set

A symbol is the smallest signalling element provided by PHY for use by the Data Link Layer. Symbols are used to convey three types of information:

a) Line states, such as the Halt Line State or Idle Line State.

b) Control sequences, such as the Starting Delimiter, Ending Delimiter or Control Indicator sequences.

c) Data quartets, each representing a group of four ordered data bits.

Peer DLL entities communicate via PHY using a set of fixed-length symbols. These symbols are passed across the PHY-to-DLL interface via the PH_UNITDATA.request and PH_UNITDATA.indication primitives. The DLL entities shall generate PDUs as matched pairs of symbols, according to the rules specified in the remainder of this subclause. These symbol pair definitions allow decoding only relative to the framing boundary established by a Starting Delimiter symbol pair.

Peer SMT entities also communicate via PHY using a set of variable-length line states. These line states are encoded as repeating symbol streams for transmission across the Physical Link; however, they can be detected without establishment of a framing boundary.

7.2.1 Line state symbols

These three symbols are used on the medium between DLL PDUs.

7.2.1.1 Quiet (Q)

The Quiet symbol indicates the absence of any transitions in the code group. This symbol shall not be transmitted or repeated by DLL entities (MAC or HRC). Detection of this symbol within a DLL PDU constitutes an error in the PDU.

7.2.1.2 Halt (H)

The Halt symbol indicates CMT signalling sequences (in the form of line states). It is also used for filtering line state or code violation symbols from the repeated symbol stream while minimizing the d.c. component of the NRZI signal on the transmission medium. This symbol shall not be transmitted or repeated by DLL entities (MAC or HRC). Detection of this symbol within a DLL PDU constitutes an error in the PDU.

7.2.1.3 Idle (I)

The Idle symbol indicates the normal condition of the medium between MAC or HRC PDUs (frames or cycles). It provides a continuous fill pattern to establish and maintain clock synchronism. The Idle symbol may also occur in CMT signalling sequences and as a fill pattern within a channel in Hybrid mode.

7.2.2 Control symbols

7.2.2.1 Starting Delimiter (SD)

A Starting Delimiter (SD) is used to delineate the starting boundary of a data transmission sequence (i.e. a MAC or HRC PDU). This PDU normally begins when the medium is in the idle condition although it may succeed or preempt a previous PDU. The SD is unique in that it may be recognized independent of previously established code group (framing) boundaries. Note that the Starting Delimiter may occur at any point regardless of the previous framing boundary. A boundary shift may occur in the transmitting PHY or, due to a clock or data recovery error, in the receiving PHY.

The symbol sequence of J followed by K, from the DLL to PHY, shall be used by the Encode function to impress the Starting Delimiter on to the transmission medium. The encoding of the JK symbol pair contains a uniquely recognizable code bit sequence that does not exist in any other legal symbol sequence regardless of previously established framing boundaries. Using this characteristic, the Decode function of the receiving PHY uses the incoming JK sequence to establish the correct framing boundary.

The DLL shall employ proper Starting Delimiter usage and legal symbol stream sequencing. PHY shall not transmit a J symbol unless it is immediately followed by a K symbol.

NOTE 6 – Legal symbol stream sequencing may be enforced by PHY, since the Repeat Filter function is located downstream from the PH_UNITDATA.request interface.

7.2.2.1.1 Initial SD symbol (J)

The J symbol is the first symbol of a sequential Starting Delimiter symbol pair (JK).

7.2.2.1.2 Final SD symbol (K)

The K symbol is the second and last symbol of a sequential Starting Delimiter symbol pair (JK).

7.2.2.2 Other delimiter symbols

7.2.2.2.1 Embedded Delimiter symbol (L)

The L symbol is the second and last symbol of a sequential Starting Delimiter symbol pair (IL) embedded within a channel in Hybrid mode. It does not affect the framing boundary because it does not contain the uniquely recognizable code bit sequence of the JK pattern. The L symbol is also used for filtering line state or code violation symbols from the repeated symbol stream in Hybrid mode.

Since the Embedded Delimiter cannot be recognized independent of symbol boundaries, proper decoding of this symbol depends on the previous establishment of the framing boundary.

The L symbol shall not be transmitted or repeated in Basic mode. It shall be filtered by the Repeat Filter according to the current mode (see 8.9).

7.2.2.2.2 Ending Delimiter (ED) symbol (T)

An Ending Delimiter symbol (T) terminates all MAC PDUs. The T symbol is not necessarily the last symbol in a transmission sequence, since the Ending Delimiter may be followed by one or more Control Indicator symbols. A sequence of Ending Delimiter and Control Indicator symbols shall be generated by the DLL as a balanced sequence of symbol pairs (i.e. an even number of R, S, and T symbols). When no Control Indicators are present, this sequence shall consist of a pair of T symbols. In Hybrid mode the T symbol is also used for filtering erroneous symbols in the Programming Template.

Note that the Ending Delimiter cannot be recognized independent of symbol boundaries, so proper decoding of this symbol depends on the previous establishment of the framing boundary. The encoding of this symbol is designed to minimize the probability that transmission errors will create a false Ending Delimiter on a stripped frame or token.

7.2.2.3 Control Indicators

Control Indicators specify logical conditions associated with a data transmission sequence. They may be independently altered by repeating nodes without altering the normal data in the transmission sequence. A sequence of Ending Delimiter and Control Indicator symbols shall be generated by the DLL as a balanced sequence of symbol pairs (i.e. an even number of R, S and T symbols). A single Ending Delimiter symbol followed by an odd number of Control Indicator symbols is a balanced symbol pair sequence; however, an Ending Delimiter symbol followed by an even number of Control Indicator symbols shall be balanced by adding a final Ending Delimiter symbol.

The encoding of the Control Indicator symbols is designed both to ensure balanced symbol pairs and to minimize the probability that transmission errors will transpose one Ending Delimiter or Control Indicator symbol into a different Ending Delimiter or Control Indicator symbol.

7.2.2.3.1 Reset symbol (R)

The Reset Symbol indicates a logical ‘off’ or ‘false’ condition.

7.2.2.3.2 Set symbol (S)

The Set Symbol indicates a logical ‘on’ or ‘true’ condition.

7.2.3 Data Quartets (0-F)

A Data Quartet symbol conveys one quartet of binary data within a data transmission sequence. The 16 Data Quartet symbols are denoted by the hexadecimal digits (0 – F), and a generic member of the set is denoted by the character ‘n’. A sequence of Data Quartet symbols shall be generated by the DLL as a balanced sequence of symbol pairs (i.e. an even number of n symbols).

7.2.4 Violation symbol (V)

The Violation symbol denotes a signal on the medium that does not conform to any legal symbol in the symbol set. The receipt of Violation symbols may result from various error conditions or during ring clock synchronization sequences. PHY shall not transmit (i.e. generate or repeat) Violation symbols.

7.3 Line states

This subclause defines the line states that determine and signal the status of a Physical Link. These line states are generated by PHY upon request from SMT (via SM_PH_LINE_STATE.request), and are detected by PHY and reported to SMT (via SM_PH_STATUS.indication). The line states represent a longer term condition of the Physical Link than that represented by a symbol or a symbol pair. Line state detection is provided at all times, but accuracy need not be guaranteed during the initial line state acquisition interval or during line state change intervals (see 8.7). Note that the following line state definitions are mutually exclusive, but not exhaustive, i.e. line conditions exist that do not satisfy the criteria for any of the defined line states. In this case, the current line state is unknown, which is the default condition.

PHY shall report any change in the received line state to SMT via the SM_PH_STATUS.indication primitive. PHY shall also report a PH_Invalid to the DLL interface whenever the received line state is QLS, HLS, MLS, or NLS.

At any time, SMT shall be able to determine the current line state via the SM_PH_CONTROL.request primitive. If the current line state is unknown, then Line State Unknown shall be reported to SMT along with the most recently known line state.

7.3.1 Quiet Line State (QLS)

A continuous stream of Quiet symbols shall be sent by PHY to signal Quiet Line State. This line state is used as part of the Connection Management process. It may also indicate the absence of a Physical Link.

Quiet Line State shall be entered upon the receipt of 16 consecutive Q symbols with Signal_Detect(on); however, entry is not required until eight consecutive QQ symbol pairs are received with the current framing boundary. Quiet Line State shall also be entered upon receipt of Signal_Detect(off) from PMD.

Quiet Line State shall be exited upon receipt of any symbol other than Q with Signal_Detect(on).

7.3.2 Halt Line State (HLS)

A continuous stream of Halt symbols shall be sent by PHY to signal Halt Line State. This line state is used as part of the Connection Management process.

Halt Line State shall be entered upon the receipt of 16 consecutive H symbols with Signal_Detect(on); however, entry is not required until eight consecutive HH symbol pairs are received with the current framing boundary.

Halt Line State shall be exited upon receipt of any symbol other than H, or upon receipt of Signal_Detect(off).

7.3.3 Master Line State (MLS)

A continuous stream of alternating Halt and Quiet symbols shall be sent by PHY to signal Master Line State. This line state is used as part of the Connection Management process.

Master Line State shall be entered upon the receipt of eight consecutive HQ (or QH) symbol pairs with Signal_Detect(on); however, entry is not required until eight consecutive HQ (or QH) symbol pairs are received with the current framing boundary.

Master Line State shall be exited upon receipt of any symbol pair other than HQ (or QH), or upon receipt of Signal_Detect(off).

7.3.4 Idle Line State (ILS)

A continuous stream of Idle symbols shall be sent by PHY to signal the Idle Line State. This line state is intended to establish and maintain clock synchronization on the outbound Physical Link. This line state is used both as part of the Connection Management process and between DLL PDU sequences during normal operation.

Idle Line State shall be entered upon the receipt of four consecutive I symbols with Signal_Detect(on) and Clock_Detect(on) (if implemented); however, entry is not required until two consecutive II symbol pairs are received with the current framing boundary. Note that this may be increased by up to 11 bits if the Elasticity Buffer function is implemented before the Line State Detection function, and the Elasticity Buffer removes the maximum number of bits permitted (i.e. 20 bits maximum to nine bits minimum equals 11 bits). In Hybrid mode, the detection of ILS shall not occur during a cycle (i.e. until a complete cycle of symbol pairs has been received since entry to ALS, except after the HRC Abort sequence (JKTT) is received).

NOTE 7 – ISO/IEC 9314-6 requires the receipt of 16 consecutive I symbols before accepting ILS during the CMT signalling process. This requirement does not apply to ILS received between DLL PDUs.

Idle Line State shall be exited upon receipt of any symbol other than I, receipt of Signal_Detect(off), or detection of Clock_Detect(off) (if implemented).

7.3.5 Active Line State (ALS)

When PHY transmits a DLL PDU sequence on the outbound Physical Link (i.e. whenever a DLL PDU sequence is sent or repeated by this node with Transmit_PDR enabled), this indicates that the Physical Link endpoint in this node is enabled, and that a DLL entity is active somewhere upstream of this Physical Link (i.e. in this node or in an upstream node). When detected, Active Line State indicates that the inbound symbol stream on the Physical Link is a DLL PDU sequence, and that the neighbouring PHY has enabled the associated Physical Link.

Active Line State shall be entered upon the receipt and acceptance of a JK symbol pair on any arbitrary code bit boundary in the input NRZ stream, with Signal_Detect(on) and Clock_Detect(on) (if implemented). Additional requirements on decoding JK and entry to ALS are specified in the clause on the Decode function (see 8.5).

Active Line State shall be exited upon receipt of any symbol other than I, n, R, S or T, receipt of Signal_Detect(off), or detection of Clock_Detect(off) (if implemented), and upon entry into Cycle Line State or Idle Line State. ALS may optionally be exited upon receipt of a mixed (data and non-data) symbol pair on the current framing boundary, receipt of an n symbol after at least one R, S, or T

symbol has been received or receipt of any symbol other than I after at least one I symbol has been received. A JK received and accepted while in ALS (on any arbitrary bit boundary) may cause both an exit from ALS and a subsequent re-entry to ALS but this is not required.

7.3.6 Cycle Line State (CLS)

When PHY transmits a cycle on the outbound Physical Link (i.e. whenever a cycle is sent or repeated by this node with Transmit_PDR enabled), this indicates that the Physical Link is enabled in this node, and that an HRC entity is active somewhere upstream of this Physical Link (i.e. in this node or in an upstream node). When detected, Cycle Line State indicates that the inbound symbol stream on the Physical Link is a cycle, and that the neighbouring PHY has enabled the associated Physical Link.

Cycle Line State shall be entered upon the receipt of a Control Indicator (R or S) symbol pair in Active Line State immediately after the JK symbol pair, with Hybrid mode implemented and enabled, Signal_Detect(on) and Clock_Detect(on) (if implemented).

Cycle Line State shall be exited upon receipt of any symbol other than I, L, n, R, S or T, receipt of any symbol other than I after the end of the cycle, receipt of Signal_Detect(off), or detection of Clock_Detect(off) (if implemented), and upon entry into Idle Line State or Active Line State. CLS may optionally be exited upon receipt of a mixed (data and non-data) symbol pair on the current framing boundary.

7.3.7 Noise Line State (NLS)

PHY shall not transmit a symbol stream that can cause Noise Line State to be detected by the neighbouring PHY. When detected, Noise Line State indicates that the inbound Physical Link is noisy and that, if NLS persists, the Physical Link is faulty.

Noise Line State shall be entered upon the occurrence of 16 or 17 potential noise events without satisfying the criteria for entry to another line state. Potential noise events shall include decoding a Q, H, J, K, or V symbol (or a symbol pair containing at least one Q, H, J, K, or V symbol) with Signal_Detect(on). An implementation may also optionally count as potential noise events:

a) An Elasticity Buffer error with Signal_Detect(on).

b) Decoding any symbol or symbol pair on the current framing boundary that could cause exit from the current or last known line state with Signal_Detect(on).

c) If Clock_Detect is implemented, decoding any symbol with Signal_Detect(on) but Clock_Detect(off).

(For a description of Signal_Detect and Clock_Detect, see 6.2.3 and 8.4, respectively.)

The count of potential noise events shall be reset to zero whenever the criteria for entering or continuing another line state are satisfied.

Noise Line State shall be exited upon satisfying the criteria for entry to another line state. If the criteria for NLS and another line state occur simultaneously, the other line state shall take precedence.

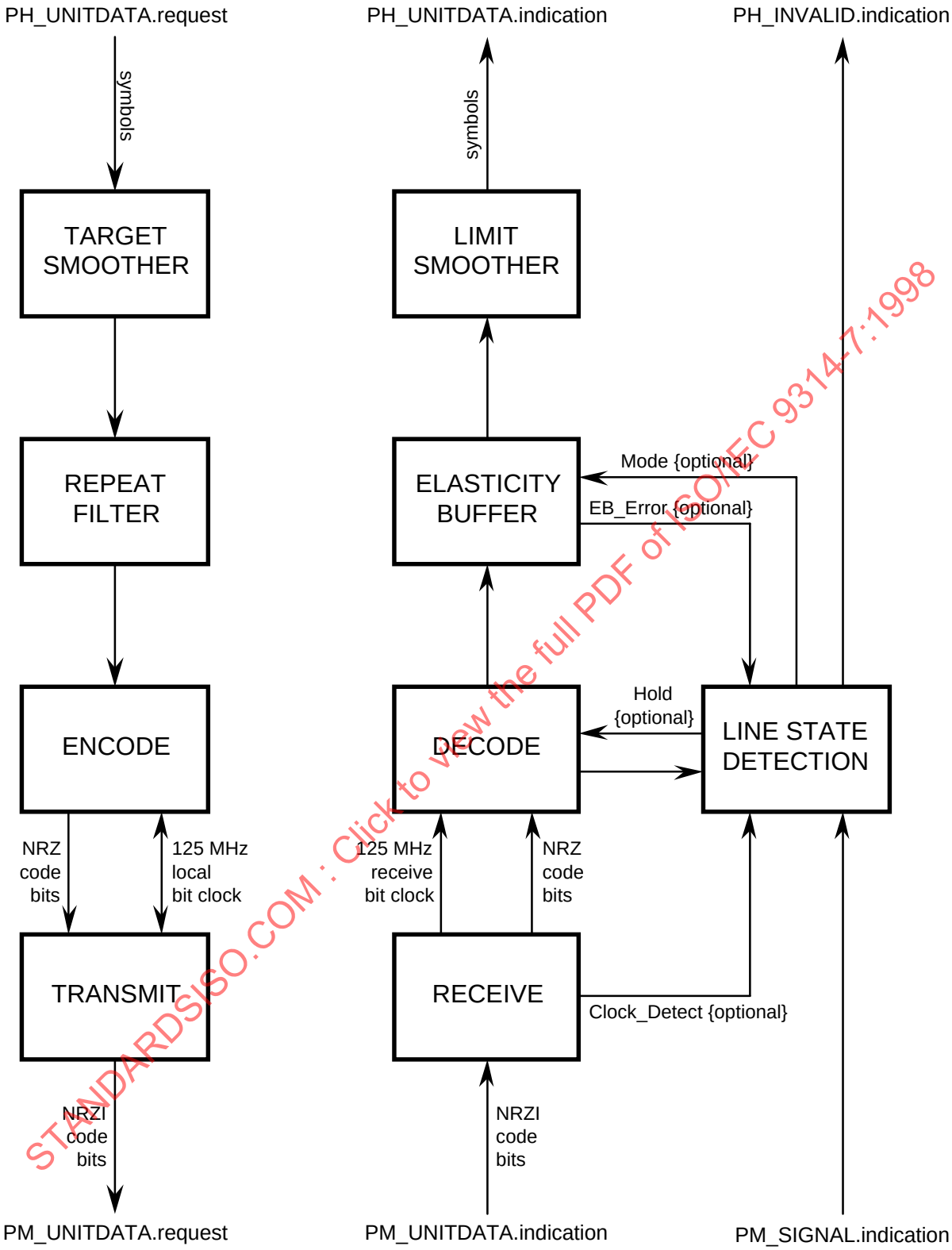


Figure 3 – PHY functional block diagram example

8 Operation

This clause specifies the operation of each of the functional components of PHY.

The functional organization of the FDDI PHY is shown in figure 3. The functional organization described in this clause is of an exemplary nature, and is not intended to specify an implementation of the PHY interfaces. Any implementation technique that causes the same external behaviour of the protocol is equally valid.

8.1 General

This subclause describes common aspects of the operation of the functional components of PHY.

8.1.1 Coding

Successful operation of a serial baseband transmission system, such as FDDI, requires the use of coding to combine the functions of data and clock transmission. Data recovery from the serial data stream requires the recovery of synchronizing clock information which is embedded in the data stream. Information is conveyed on the transmission medium by impressing transitions, or lack of transitions, on the medium. The minimum time interval between possible transitions on the interface medium is defined as the 'code cell.' Ideally, each transition, or lack of a transition, represents a useful piece of data. However, in practice, this is not feasible because an extended series of bits represented by no transition would not contain sufficient information to recover the synchronizing clock. Furthermore, for high-speed serial transmission, it is desirable that d.c. balance be maintained to the degree feasible in order to facilitate interface component and circuit designs.

The FDDI PHY employs a dual embedded coding structure so as to achieve these characteristics. The resultant serial code bit stream, as seen on the transmission medium, contains at least two transitions for each transmitted data symbol and is thus self clocking, has a maximum of three consecutive code cell zeros for each transmitted data symbol and is thus run-bounded, and yields a ±10 % maximum cumulative d.c. component variation from nominal centre for all valid symbol sequences.

In FDDI, information is signalled as a stream of fixed-length code groups, each containing five code bits. Each valid code group represents a symbol. The first level of coding performed by PHY is the conversion of symbols from the Data Link Layer (MAC or HRC) to code groups. Each symbol is mapped to its corresponding code group, consisting of a sequence of five NRZ code bits. The second level of coding performed by PHY is the conversion of NRZ code bits to NRZI code bits. Each NRZ code bit 'one' generates a transition in the polarity of the NRZI signal, while an NRZ code bit 'zero' maintains the previous NRZI signal polarity. For the incoming pulse stream, NRZI code bits are first decoded to NRZ code bits and then decoded to symbols for delivery to the DLL.

The interface between the DLL and PHY uses symbols to convey logical meaning. PHY presents a symbol to the DLL and accepts a symbol from the DLL once every symbol time. Since one symbol maps to one five-bit code group, a symbol time is five times the code cell time. The transmitted sequence of these symbols is determined by the DLL. The DLL shall ensure that the sequence of symbols presented to PHY for coding and transmission onto the medium, is valid and in conformance with the rules of symbol sequencing as further defined in clause 7.

The mapping between symbols and code groups is defined in table 1. The rules for usage of symbols, and the meanings assigned to these symbols, are defined in clause 7.

Table 1 defines invalid code groups that shall not be transmitted on to the medium because they cause an unacceptable d.c. component in the a.c. signal or they cause an unacceptable number of consecutive zeros on the transmission medium. PHY shall indicate the receipt of an invalid code group to the DLL as a Violation (V) or (optionally) a Halt (H) symbol.

8.1.2 Clocking

A local clock is used to synchronize both the internal operation of PHY and its interface to the Data Link Layer. This clock shall be derived from a fixed frequency reference. This reference may be created internally within the PHY implementation or supplied to PHY. (A crystal oscillator may be used for this purpose.)

Characteristics of the local clock shall be:

a) Nominal symbol time (UI) = 40 ns (1/UI = 25 MHz)

b) Nominal code bit cell time (UI) = 8 ns (1/UI = 125 MHz)

c) Frequency accuracy < ±0,005 % (±50 ppm)

d) Harmonic content (above 125,02 MHz) < –20 dB

e) Phase jitter (above 20 kHz) < ±8° (0,044 UI pp)

f) Phase jitter (below 20 kHz in Hybrid mode) < ±270° (1,5 UI pp)

The Receive function derives a clock by recovering the timing information from the incoming serial bit stream. This clock is locked in frequency and phase to the transmit clock of the upstream node. The maximum difference between the received bit frequency and the local bit frequency is 0,01% of the nominal frequency. The received frequency can be either slower or faster than the local frequency, resulting in an excess or a deficiency of bits unless some compensation is included. The Elasticity Buffer function provides this compensation by adding or dropping Idle bits in the preamble between DLL PDUs.

The operation of the Elasticity Buffer function produces variations in the lengths of the preambles between DLL PDUs as they circulate around the logical ring. The cumulative effect on preamble size of PDU propagation through many Elasticity Buffers can result in excessive preamble erosion and, in Hybrid mode, excessive cycle clock jitter. The Smoothing function serves to filter out these undesirable effects (see 8.8).

Figure 7 shows the relationship between the FDDI jitter tolerances in Hybrid mode and the standard jitter tolerances specified by the public network.

8.1.3 Latency

This subclause specifies the latency requirements for an FDDI node. Node delay is specified as the (MIC to MIC) propagation delay of a Starting Delimiter through a node. This includes the total delay of the repeat path from the optical input interface of the MIC where the Starting Delimiter enters an active node, to the optical output interface of the MIC where the Starting Delimiter next leaves that node. This may be the same MIC or a different MIC, depending upon the internal configuration of the node, (e.g. a Dual Attachment Station may be in either a Wrap or a Thru configuration, as defined in the part of ISO/IEC 9314 on SMT). The node delay may be different for different MICs and different internal configurations of the same node.

Ring latency is the cumulative effect of alternating node delays and cable plant delays around the logical ring (as configured). Annex A provides information for calculating maximum ring latency.

In the special case of a two-node ring composed of two PHYs (one in each node) and one MAC, a minimum ring latency of five bytes (symbol pairs) is required to guarantee adequate preamble on a circulating token when the MAC is in repeat mode. This preamble is needed to ensure the proper operation of certain permitted implementations of the Elasticity Buffer function in each PHY, while maintaining the integrity of the circulating token. To ensure interoperability in this case, an implementation shall guarantee a minimum latency of three bytes when a MAC is configured in a repeat path through a node, and a minimum of two bytes when a MAC is not configured in a repeat path through a node.

ISO/IEC 9314-2, ISO/IEC 9314-5 and ISO/IEC 9314-6 contain timers whose values are dependent on a deterministic upper bound on ring latency. To provide an orderly method for sizing FDDI networks and setting timers, an implementation shall not exceed the following node delays:

a) Node delay shall not exceed 1,000 µs per port when the optional HRC function is not present in Basic mode, and the Starting Delimiter is preceded by an even number of symbols (since the previous Starting Delimiter) and by sufficient preamble to recentre the Elasticity Buffer and empty the smoothers.

b) Node delay shall not exceed 1,480 µs per port when the optional HRC function is present in Basic mode, and the Starting Delimiter is preceded by an even number of symbols (since the previous Starting Delimiter) and by sufficient preamble to recentre the Elasticity Buffer and empty the smoothers.

c) Node delay shall not exceed 1,720 µs per port when the optional HRC function is present in Hybrid mode, no Latency Adjustment Buffer is configured in the repeat path, and the Starting Delimiter is preceded by an integral number of symbols (since the previous Starting Delimiter) and by an appropriate sequence of cycles and preambles to recentre both the Elasticity Buffer and the Limit Smoother.

These node delay requirements are determined based upon the total number of ports that can be configured in a given repeat path through the station, rather than the subset of those ports that are currently configured in the repeat path (e.g. internally bypassed ports in a station or concentrator may still contribute to repeat path delay).

NOTE 8 – To ensure compliance with these node delay requirements, an implementation should be designed with a minimum node delay that is less than the specified delay for each category by a sufficient amount to allow for internal quantizing errors within PHY. Quantizing errors occur both in synchronization of the incoming signal stream to the receive bit clock, and in synchronizing decoded symbols to the local symbol or byte clock. A safe margin for quantizing errors would be about ±84 ns for a byte-wide implementation, yielding a design delay budget of 0,916 µs for rule (a), 1,396 µs for rule (b) and 1,636 µs for rule (c). For rule (c) the Target Smoother introduces an additional potential quantizing error of ±1 symbol; the design budget of 1,636 µs must cover the case where the Target Smoother is full.

Annex A provides information for calculating maximum ring latency.

8.2 Encode function

The Encode function encodes the symbol stream derived from the current Line_State_action or the PH_UNITDATA.request, into an equivalent NRZ code bit stream for presentation to the Transmit function. The Encode function is controlled by SMT via the SM_PH_LINE_STATE.request. When the current Line_State_action is other than Transmit_PDR, the Encode function shall ignore the PH_UNITDATA.request and continuously encode the symbols generated by the Line_State_action.

The Encode function shall encode each symbol into a unique five-bit code group as defined in table 1. Each code group shall be presented to the Transmit function as a serial stream of NRZ code bits.

The Encode function shall use the local clock to encode the symbol stream into code groups and to present the code bits to the Transmit function.

8.3 Transmit function

The Transmit function is responsible for encoding the NRZ serial code bit stream from the Encode function into an equivalent NRZI pulse stream for presentation to PMD. An annex to the part of ISO/IEC 9314 on PMD provides an exemplary interface specification.

The Transmit function shall use the local clock to generate the NRZI pulse stream.

8.4 Receive function

The Receive function is responsible for decoding the NRZI pulse stream received from PMD into an equivalent NRZ pulse stream for presentation to other functions within PHY. An annex to the part of ISO/IEC 9314 on PMD provides an exemplary interface specification.

The Receive function shall derive a clock, at the code bit frequency (125 MHz), from the incoming NRZI pulse stream. (A phase-locked loop may be used for this purpose.) This clock shall be used to decode the NRZI pulse stream during the Idle, Active and Cycle Line States, and may also be used to synchronize detection of the Halt and Master Line States. Alternatively, some other synchronization technique may be used to detect the Halt and Master Line States.

The Receive function may provide an optional Clock_Detect signal that, when asserted, indicates that the derived clock is successfully locked in frequency and phase to the incoming NRZI pulse stream. When implemented, the Clock_Detect signal is used by the Line State Detection function.

Receiver clock acquisition time is constrained by the line state detection times specified in 8.7.

8.5 Decode function

The Decode function decodes the serial NRZ code bit stream, synchronized to the receive clock, into an equivalent symbol stream. The Decode function establishes code group framing boundaries and maintains symbol (or byte) synchronization as appropriate to the implementation and the mode of operation.

The Decode function shall decode the incoming NRZ code bit stream into a corresponding symbol stream as defined in table 1.

The Decode function shall recognize a Starting Delimiter when:

a) Signal_Detect(on) is asserted; and

b) Clock_Detect is not implemented or Clock_Detect(on) is asserted; and

c) 1) an error-free JK code bit pattern is received on any bit boundary, and the current or last known line state is ILS; or

2) an error-free JK code bit pattern is received in Basic mode on a current symbol (5-bit) boundary, the current or last known line state is ALS, at least four symbols have been received since ALS was last entered, and at least one II symbol pair has been received on the current byte (10-bit) boundary since ALS was last entered; or

3) an error-free JK code bit pattern is received in Hybrid mode on a current symbol (5-bit) boundary, the current or last known line state is ALS or CLS, and at least four symbols have been received since ALS was last entered.

The Decode function may optionally recognize a Starting Delimiter when:

d) Signal_Detect(on) is asserted; and

e) Clock_Detect is not implemented or Clock_Detect(on) is asserted; and

f) 1) an error-free JK code bit pattern is received on any bit boundary, and either the current or last known line state is not CLS or CLS was not entered via an error-free JKS symbol sequence or the last known line state was CLS and a previous error-free JK code bit pattern was received after entering CLS; or

2) an error-free JK code bit pattern is received on a current symbol (5-bit) boundary; or

3) a JK code bit pattern is received on a current symbol (5-bit) boundary with a pair of interior error bits (i.e. a pair of NRZ code errors within the JK pattern that could be caused by a single error in the NRZI waveform) or one edge error bit (i.e. an NRZ code error at the beginning or end of the JK pattern that could be caused by a single error in the NRZI waveform), and the current or last known line state is ILS; or

4) a JK code bit pattern is received on a current symbol (5-bit) boundary with one error symbol, and the current known line state is ILS (i.e. either the symbol sequence (I, I, I, I, J, ~(I or K)) or the symbol sequence I, I, I, I, ~(I or J), K) is received).

The Decode function shall present any Starting Delimiter pattern recognized according to these rules as a JK symbol pair, and the framing boundary shall be adjusted if necessary. Any other received J symbol or JK symbol pair shall be presented as one or more violation (V or H) symbols and the framing boundary shall not be adjusted.

In practice, the Decode function may be required to insert a variable 0 to 4 code bit delay (or a 0 to 9 code bit delay for byte-wide implementations) into the data stream to allow the symbol (or byte) clock to maintain a constant phase when the code group framing boundary changes (due to the recognition of a new Starting Delimiter sequence). If the Decode function adds or deletes code bits from the data stream when adjusting code group boundaries, then the combined effect of the Elasticity Buffer function and the Decode function shall follow the rules for adding and deleting bits specified for the Elasticity Buffer function (see 8.6).

8.6 Elasticity Buffer function

An Elasticity Buffer shall be provided in each node to compensate for the difference in frequencies between the receive clock and the local clock. An elasticity buffer is similar in function to a first-in first-out memory, which is filled halfway before bits are removed. Data is written into the Elasticity Buffer function using the receive clock. Data is read from the elasticity buffer function using the local clock.

The minimum required elasticity shall be greater than ±4,5 code bits. The required elasticity is calculated as follows: 9 000 symbols equals 45 000 code bits. With a clock tolerance of 0,005 % the maximum frequency variance between the receive clock and the local clock is 0,01 %. Calculating 0,01 % of 45 000 code bits yields 4,5 code bits.

To allow for bits that are to be dropped when the local frequency is less than the receive frequency, the Data Link Layer entity (MAC or PHY) inserts a preamble of Idle symbols between DLL PDUs. In Basic mode, MAC shall insert at least 16 Idle symbols between each MAC PDU (frame or token) to be transmitted. In Hybrid mode, HRC shall insert at least four Idle symbols between each HRC PDU (cycle) to be transmitted. The operation of the Elasticity Buffer in subsequent repeating nodes may change the length of the preamble as described herein.

Although figure 3 shows the Elasticity Buffer function after the Decode function, any implementation that meets the following rules is permitted:

a) When entering Active Line State from Idle Line State and both the receive clock and the local clock are within tolerance, the initial JK symbol pair and the subsequent contiguous input symbol stream shall be reproduced from the input NRZ bit stream using the symbol framing reference provided by the Starting Delimiter pattern, without inserting, deleting or modifying any symbols, until one of the following occurs:

   1) At least 9 000 PH_UNITDATA.indication symbols have been presented since the last entry to Active Line State (see rule (b)); or

   2) At least one Idle symbol has been presented in Basic mode, or a complete cycle of symbols has been presented in Hybrid mode, or an HRC Abort sequence (JKTT) has been presented; or

   3) Zero to nine code bits (depending on the difference between the new and previous framing boundaries) prior to presentation of another JK symbol pair recognized as a Starting Delimiter pattern (see rule (e) and 8.5); or

   4) A symbol is presented that causes an exit from Active Line State in Basic mode (see rule (f)).

   NOTE 9 – If the Repeat Filter or Smoothing functions are implemented prior to the PH_UNITDATA.indication, their operation may also modify the PH_UNITDATA.indication from that represented by the input NRZ stream.

b) When in Active Line State or Hybrid mode and error conditions exist in the local or upstream node (e.g. if either the receive clock or the local clock are out of tolerance, or after 9 000 consecutive symbols have been presented without reaching condition (a)(2)), an Elasticity Buffer error may occur, after which symbols may be inserted, deleted or modified from the input NRZ bit stream. All Elasticity Buffer errors while in Active Line State or Hybrid mode shall be reported to SMT, and shall also be reported to the DLL as one or both of the following:

   1) A PH_UNITDATA.indication with a PH_Indication(V or H) symbol parameter; or

   2) A PH_INVALID.indication. This indication is required in Hybrid mode and is recommended in Basic mode.

c) After satisfying rule (a)(2), an implementation may insert or delete Idle symbols or bits in the received NRZ bit stream without causing an Elasticity Buffer error.

   After entering Quiet, Halt, Master or Noise Line State, an implementation may insert or delete Quiet, Halt, Idle or Violation symbols or bits in the received NRZ bit stream without causing an Elasticity Buffer error.

d) While in Idle Line State, an implementation shall be capable of inserting or deleting Idle symbols or bits in the received NRZ bit stream without causing an Elasticity Buffer error.

e) When the current or last known line state is Active or Cycle Line State, if another Starting Delimiter pattern is recognized (see 8.5) on any arbitrary boundary in the input NRZ bit stream, up to four code bits may be inserted or deleted (nine code bits for a byte-wide implementation) in the NRZ code bit stream prior to presenting the JK symbol pair. In this case, the requirement of at least 9 000 PH_UNITDATA.indication symbols before an Elasticity Buffer error is allowed is calculated from the last point of entry to Active Line State from Idle Line State.

If additional symbols are presented, either they shall be Idle symbols or they shall be the decoded symbols representing the duplicated bits of the received code bit stream, possibly including from one to nine leading bits of the JK symbol pair.

Deleted data shall only be those code bits between the new and previous framing boundaries except as otherwise permitted by rule (a).

An implementation shall not present a J not followed by a K, nor shall it present a JK symbol pair that is not recognized as a Starting Delimiter by the Decode function (see 8.5).

f) When not in the Active or Idle Line State or in Hybrid mode, the presented symbol stream can be altered from the input NRZ bit stream; however, upon exit from Active Line State, at least the first four consecutive invalid symbols (i.e. neither Idle symbols nor JK symbol pairs) shall be propagated as invalid symbols. Subsequently, Idle symbols may be generated, but spurious J symbols or JK symbol pairs shall not be generated. Elasticity Buffer errors may occur, but only need to be recognized to the extent that they contribute to Noise Line State detection.

8.7 Line State Detection function

The Line State Detection function is used to determine the line state of the inbound Physical Link. The line states represent a longer term condition of the Physical Link than that represented by a symbol or a symbol pair (see 7.3 for the definitions of the line states). The Line State Detection function also uses the PM_SIGNAL.indication, the optional Clock_Detect signal from the Receive function, and the Elasticity Buffer error signal, to determine the current line state.

The Line State Detection function shall notify the local SMT entity (via SM_PH_STATUS.indication) of any changes to the detected line state (see 6.3.2). This function shall also notify the local DLL entity (via PH_INVALID.indication) when a PH_Invalid condition is detected (see 6.1.3).

The initial line state detection interval begins when both a Signal_Detect(on) and a PM_Indication(NRZI code) serial data stream meeting the criteria for entering or maintaining the Halt, Master or Idle Line State are received from PMD, and continues until the line state reported to SMT correctly indicates the state of the serial data stream being received from PMD. This interval shall not exceed the maximum PHY acquisition time (AT_Max). During this interval Line State Unknown or Noise Line State, as appropriate, shall be reported to SMT (see 7.3). The default value of AT_Max shall be 100 µs. AT_Max shall not exceed 100 µs.

After initial line state detection, the correct line state may change or be temporarily lost. Loss of the correct line state results from internal conditions within the receiving PHY entity (e.g. loss of receive clock synchronization or Elasticity Buffer errors), whereas changes in line state result from changes outside the receiving PHY entity (e.g. change in transmitted line state or in the state of PMD). The line state change detection interval begins when the current or last known line state is not Quiet Line State and either Signal_Detect(off) or both Signal_Detect(on) and a PM_Indication(NRZI code) serial data stream meeting the criteria for entering or maintaining the Quiet, Halt, Master or Idle Line State are received from PMD, and continues until the line state reported to SMT correctly indicates the state of the serial data stream being received from PMD. This interval shall not exceed the maximum line state change interval (LS_Max). During this interval Line State Unknown or Noise Line State, as appropriate, shall be reported to SMT (see 7.3). The default value of LS_Max shall be 25 µs. LS_Max shall not exceed 25 µs.

Entry to Active or Cycle Line State, and return to Idle Line State when the current or last known line state is Idle, Active or Cycle Line State, shall be detected immediately and reported to SMT (see 7.3).

Note that the detection of the Idle, Active and Cycle Line States (and optionally the Halt and Master Line States) requires receive clock synchronization. Consequently, receive clock acquisition time in an implementation is constrained by the AT_Max or the LS_Max criteria, or by both. If an implementation requires receive clock synchronization to detect Halt or Master Line State, then receive clock acquisition time on these patterns (HLS or MLS) must be less than AT_Max. Otherwise, a subsequent switch from Halt or Master Line State to Idle or Active Line State requires receive clock acquisition on these patterns (ILS or ALS) in less than LS_Max.

The Line State Detection function shall optionally detect entry to Hybrid mode and return to Basic mode. Basic mode shall be entered upon detection of any of the following conditions:

a) PHY_Reset is requested; or

b) Disable_Hybrid_mode is asserted; or

c) PH_Invalid is detected (return to Basic mode is not required if PH_Invalid was caused solely by an Elasticity Buffer error or other internal error); or

d) The current line state is Active Line State, the first symbol received after the JK symbol pair is an 8 or C data quartet, and the second symbol received after the JK symbol pair is any data quartet. (This is the start of frame sequence for a MAC recovery frame or token.)

Hybrid mode shall be entered when the optional Cycle Line State is entered.

8.8 Smoothing function

Each PHY shall process the symbol stream using a Smoothing function. This function compensates for the cumulative effect of multiple PHY Elasticity Buffer functions deleting or adding symbols to the same preamble. Unconstrained preamble erosion can result in loss of frames. In Hybrid mode, the preamble length must be carefully controlled to limit jitter in the cycle clocking function. The Smoothing function absorbs surplus symbols from longer preambles and redistributes them into shorter preambles. This significantly reduces the variance of preamble sizes during long bursts of frames or cycles.

The Basic mode design considerations include:

a) The Elasticity Buffer is not required to recentre on preambles shorter than four symbols;

b) MAC is not required to repeat frames with preambles shorter than two symbols;

c) MAC is not required to copy frames with preambles shorter than 12 symbols.

In Basic mode, the Smoothing function shall be capable of inserting at least two preamble symbols into *repeated* preambles shorter than 14 symbols. In stations whose MACs require from nine to 12 preamble symbols to copy frames properly, the Smoothing function shall be capable of inserting at least two preamble symbols into *received* preambles shorter than 14 symbols. This smoothing capability shall be reclaimed by deleting excess symbols from preambles longer than 14 symbols. In stations whose MACs require 11 or 12 preamble symbols to copy frames properly, the Smoothing function shall also be capable of inserting at least two additional preamble symbols into *received* preambles shorter than 12 symbols. This additional smoothing capability shall be reclaimed by deleting excess symbols from preambles longer than 12 symbols.

The Smoothing function shall be capable of reclaiming additional space from stripped partial frames in Basic mode. This space may be reclaimed by deleting frame symbols or by replacing them with Idles. In addition, the Smoothing function may optionally reclaim space from other partial frames, provided that format errors are not lost. A format error is not lost if, either it is counted in PHY and reported to SMT, or it is correctly propagated to the next MAC or Repeat Filter function. After a token, partial SDU (stripped SDU or format error) or HRC abort sequence (JKTT), the Smoothing function may optionally delete excess symbols from preambles longer than four symbols.

The Hybrid mode design considerations include:

d) The Elasticity Buffer is not required to recentre on preambles shorter than four symbols;

e) HRC is not required to synchronize cycles with preambles longer than six symbols;

f) 8 kHz cycle clock jitter must be minimized.

In Hybrid mode, the Smoothing function shall be capable of inserting and deleting at least two preamble symbols into or from *received* preambles shorter than four symbols and longer than six symbols, respectively. This smoothing capability shall be reclaimed by deleting or inserting excess symbols from or into preambles longer than or shorter than five symbols, respectively. The Smoothing function shall also be capable of inserting and deleting two or four preamble symbols into or from *repeated* preambles shorter or longer than five symbols, respectively. In a byte-wide implementation, this additional smoothing capability shall operate on *transmitted* preambles. This additional smoothing capability shall be reclaimed by deleting or inserting excess symbols from or into preambles longer than or shorter than five symbols, respectively.

The required smoothing capabilities shall be located somewhere after the Elasticity Buffer function in each repeat path through a node. In Basic mode, if a MAC requiring nine or more preamble symbols to copy frames properly is configured in a repeat path, then the total required smoothing capacity shall be located between the Elasticity Buffer function and the MAC Receiver in that path. For a byte-wide implementation in Hybrid mode, the required smoothing capability at the five-symbol threshold shall be located after the PH_UNITDATA.request interface in each repeat path.

If the Smoothing function is implemented before line state detection, then it shall not:

g) cause improper detection of Idle Line State as a result of inserting Idle symbols when not in Idle or Active Line State or Hybrid mode;

h) prevent proper detection of Quiet, Halt, Master or Noise Line State, or Line State Unknown as a result of deleting symbols that are potential noise events.

Given that a preamble consists of Idle symbols in the absence of noise, it is possible to relax some constraints on preamble processing with minimal impact on reliability. Specifically, during preamble processing in Basic mode the Smoothing function is not required to:

i) insert Idle symbols into a preamble except after four consecutive Idle symbols are received;

j) delete non-Idle symbols from a preamble.

An implementation is permitted to maintain its counters and to adjust smoother extension and preamble length in quantum units of bits, symbols, or bytes (symbol pairs). This implies a corresponding restriction on the elasticity buffer, i.e. the elasticity buffer quantum shall not be larger than the maximum permitted Limit Smoother quantum (one byte).

8.8.1 Limit Smoother

The Limit Smoother is required in Basic mode if a MAC requiring nine or more preamble symbols to copy frames properly is configured in the repeat path, or if the optional Target Smoother is not configured. Otherwise, the Limit Smoother is optional in Basic mode.

The Limit Smoother is required in Hybrid mode.

Figure 4 shows the Limit Smoother function expressed as a state diagram. In the state diagram, states are shown as vertical staffs and state transitions as horizontal arrows, with the triggering event or condition above the shaft and any action beneath the shaft.

This state machine defines the operation of the Smoothing function implemented immediately before the PH_UNITDATA.indication interface. Any implementation that is interoperable with this state machine is permitted unless otherwise prohibited by this part of ISO/IEC 9314.

The state machine uses the following parameters:

a) **Limit_max:** The maximum Limit Smoother capacity (in symbols). In stations whose MACs require 11 or 12 preamble symbols to copy frames properly, or in Hybrid mode, Limit_max shall be at least four symbols (two bytes); otherwise, Limit_max shall be at least two symbols (one byte). In Hybrid mode Limit_max shall be an even number of symbols;

b) **Limit_cntr:** The Limit Smoother centre point (in symbols). Limit_cntr shall be at least two symbols. In stations whose MACs require 11 or 12 preamble symbols to copy frames properly, Limit_cntr shall not exceed Limit_max minus two symbols. In Hybrid mode Limit_cntr shall be one half of Limit_max.

The state machine uses the following variables:

a) **Limit_ct:** The current Limit Smoother extension (in symbols). Byte-wide implementations are optionally permitted to count symbol pairs, rather than symbols (i.e. odd symbols can be ignored);

b) **Out_ct:** The number of symbols output in the current state. Byte-wide nodes are optionally permitted to count symbol pairs, rather than symbols (i.e. odd symbols can be ignored);

c) **PA_max:** The maximum number of preamble symbols that can be received in Hybrid mode before the next cycle is considered unsynchronized. At the input to the Limit Smoother, PA_max may be implemented as any value within the range:

(6 symbols + Elasticity Buffer quantum) ≤ PA_max ≤ 10 symbols:

d) **C_Flag:** Indicates that the current SDU is being processed as a cycle in Hybrid mode;

e) **S_Flag:** Indicates that the current or next preamble should be smoothed in Hybrid mode.

f) **T_Flag:** Indicates that the current SDU should not be stripped in Basic mode.

8.8.1.1 State LS0: Preamble (PA)

In this state the Smoother function is processing preamble symbols. The smoother contracts when excess preamble symbols beyond the thresholds are processed. Out_Ct counts the number of output preamble symbols for threshold comparison. For interoperability, the smoother is not required to delete symbols in the exact sequence described by the PA_Actions, nor to delete non-Idle symbols in Basic mode, provided that the counters are accurate upon exit from the PA state. If the smoother contracts at the 12-symbol threshold in Basic mode, it is permitted, but not required, to also contract at the 14-symbol threshold during the same preamble.

a) **LS(00): Reset:** When a PHY_Reset signal is received or optionally when PH_Invalid is detected as input, Reset_Actions shall be performed. Reset_Actions initializes the smoother.

b) **LS(01): Start of SDU:** When a Starting Delimiter (JK) symbol pair is detected as input, Start_Actions shall be performed and a transition to State LS1 shall occur. Start_Actions attempts to extend short preambles by inserting Idle symbols, with a corresponding expansion of the smoother. A symbol-wide implementation is permitted to trigger this transition when a J symbol is detected as input; however, the J symbol shall not be transmitted unless it is immediately followed by a K symbol. In Basic mode, an implementation is permitted to use any threshold value between four and 14 symbols when Start_Actions is invoked after a token or partial SDU (stripped SDU or format error). On entry to Hybrid mode, the Limit Smoother shall be recentred (and the S_Flag set) either before or after the first Cycle (the state machine shows this action occurring after the first Cycle).

8.8.1.2 State LS1: Service_Data_Unit (SDU)

In this state the Smoother function is processing SDU (frame, token or cycle) symbols. The smoother outputs all input symbols in this state. The C_Flag shall be set on entry to Hybrid mode and cleared on return to Basic mode. The T_Flag shall be set by the first occurrence of a non-data symbol in Basic mode, and may optionally be set when the HRC abort sequence (JKTT) is detected. Out_Ct shall count the number of output SDU symbols, to ensure that End_Actions does not delete symbols prior to the start of a Basic mode SDU, or the end of a Hybrid mode SDU (cycle).

a) **LS(10a): End of SDU:** A transition to State LS0 shall occur when an Idle (I) symbol is detected as input in Basic mode or while the T_Flag is set, or at the end of a cycle in Hybrid mode. If neither the C_Flag nor the T_Flag is set, indicating a stripped Basic mode SDU, then End_Actions shall be performed to reclaim space by deleting previous SDU output symbols and contracting the smoother, or by replacing them with Idles without contracting the smoother. A byte-wide implementation shall trigger this transition in Basic mode when an Idle is detected as the first symbol of a symbol pair, and is optionally permitted to trigger this transition after an Idle is processed as the second symbol of a symbol pair. This transition may optionally occur for other input conditions that terminate the SDU (e.g. in Basic mode, format errors or end of FS field; or end of HRC abort sequence (JKTT) in either Basic or Hybrid mode), provided that End_Actions does not lose completed SDUs or format errors.

b) **LS(10b): Reset:** A transition to State LS0 shall occur when a PHY_Reset signal is received or optionally when PH_Invalid is detected as input, and Reset_Actions shall be performed. Reset_Actions initializes the smoother.

c) **LS(11): Start of SDU:** When a Starting Delimiter (JK) symbol pair is detected as input, the Abort_Actions shall be performed to recentre the Limit Smoother if in Hybrid mode (by moving Limit_ct toward Limit_cntr if C_Flag is set) and a transition to State LS1 shall occur. No preamble adjustment is required on this transition in Basic mode or in implementations that set the T_Flag when the HRC Abort sequence (JKTT) is detected. A symbol-wide implementation is permitted to trigger this transition when a J symbol is detected as input; however, the J symbol shall not be transmitted unless it is immediately followed by a K symbol.

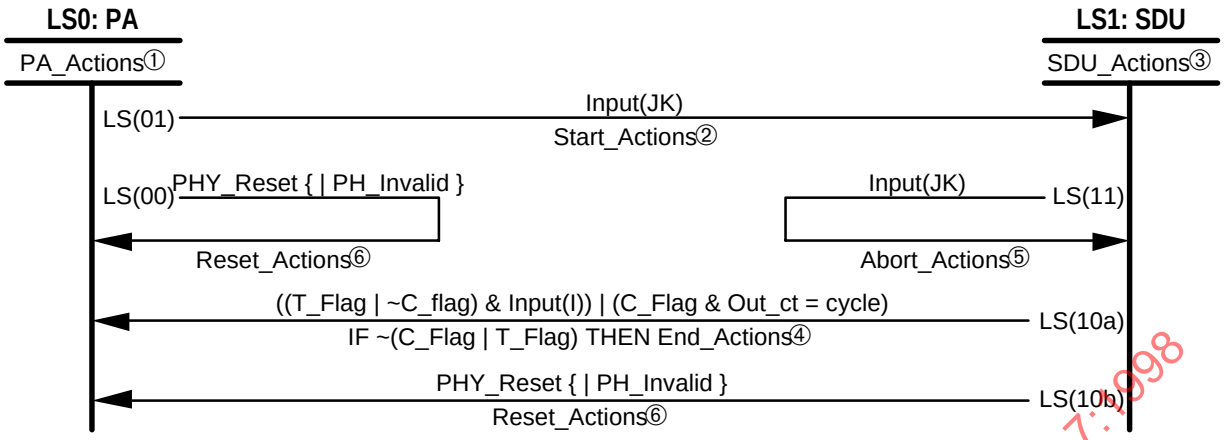


Figure 4 – Limit Smoother state diagram (Part 1 of 2)

1. PA_Actions:

```
ON entry DO
      CLEAR Out_ct
FOR EACH Input(symbol) DO
      IF { (C_Flag & Limit_ct > Limit_cntr & Out_ct = 4) |
            (C_Flag & Limit_ct > 0 & Out_ct = 6) | }
            ( { ~C_Flag & } Limit_ct > Limit_cntr & Out_ct = 12) |
            ( { ~C_Flag & } Limit_ct > 0 & Out_ct = 14)
            THEN DEC Limit_ct
            ELSE Output(I); INC Out_ct
```

2. Start_Actions:

```
{ IF Out_ct > PA_max
      THEN CLEAR S_Flag }
WHILE { (C_Flag & Limit_ct < Limit_cntr & (~S_Flag | Out_ct < 6)) |
            (S_Flag & Limit_ct < Limit_max & Out_ct < 4) | }
            ( { ~C_Flag & } Limit_ct < Limit_cntr & Out_ct < 14) |
            ( { ~C_Flag & } Limit_ct < Limit_max & Out_ct < 12)
      DO INC Limit_ct; Output(I); INC Out_ct
{ SET S_Flag = C_Flag }
```

3. SDU_Actions:

```
ON entry DO
      CLEAR T_Flag, Out_ct
FOR EACH Input(symbol) DO
      { IF Out_ct = 3
            THEN IF prev = (R | S) & Input(R | S)
                        THEN SET C_Flag
                  IF prev = (8 | C) & Input(n)
                        THEN CLEAR C_Flag, S_Flag
                  { IF prev = T & Input(T)
                        THEN SET T_Flag; CLEAR S_Flag }
      SET prev = Input(symbol); }
      IF Input(~n) { & ~C_Flag & (Out_ct > 3 | ~Input(R or S)) }
            THEN SET T_Flag
      Output(Input(symbol)); INC Out_ct
```

4. End_Actions:

```
WHILE Limit_ct > 0 & Out_ct > 0
      DO DEC Limit_ct, Out_ct
```

5. Abort_Actions:

```
{CLEAR S_Flag;
{ WHILE C_Flag & Limit_ct > Limit_cntr & Out_ct > 0
      DO DEC Limit_ct, Out_ct
  WHILE C_Flag & Limit_ct < Limit_cntr
      DO INC Limit_ct; Output(I); INC Out_ct
IF C_Flag
      THEN SET S_Flag } }
```

6. Reset_Actions:

```
CLEAR { C_Flag, S_Flag, } Limit_ct
```

Figure 4 – Limit Smoother state diagram (Part 2 of 2)

8.8.2 Target Smoother

The Target Smoother is required in Basic mode if the optional Limit Smoother is not configured. Otherwise, the Target Smoother is optional in Basic mode.

The Target Smoother is required in Hybrid mode.

Figure 5 shows the Target Smoother function expressed as a state diagram. In the state diagram, states are shown as vertical staffs and state transitions as horizontal arrows, with the triggering event or condition above the shaft and any action beneath the shaft.

This state machine defines the operation of the Smoothing function implemented immediately after the PH_UNITDATA.request interface. Any implementation that is interoperable with this state machine is permitted unless otherwise prohibited by this part of ISO/IEC 9314.

The state machine uses the following parameters:

a) **Target_max:** The maximum Target Smoother capacity (in symbols). In Basic mode Target_max shall be at least two symbols (one byte). In Hybrid mode Target_max shall be either two symbols (one byte) or four symbols (two bytes) to avoid excessive damping in the filter transfer function;

b) **Target_cntr:** The Target Smoother centre point (in symbols) in Hybrid mode. In Hybrid mode Target_cntr shall be one half of Target_max .

The state machine uses the following variables:

a) **Target_ct:** The current Target Smoother extension (in symbols). Byte-wide implementations are optionally permitted to count symbol pairs, rather than symbols (i.e. odd symbols can be ignored);

b) **Target_th:** The current Target Smoother threshold (in symbols). In Basic mode, Target_th shall be exactly 14 symbols, except that an implementation is permitted to use any value between four and 14 symbols after a token, partial SDU (stripped SDU or format error) or HRC abort sequence (JKTT). In Hybrid mode, Target_th shall be five symbols; except that byte-wide implementations are optionally permitted to alternate between four and six symbols;

c) **Out_ct:** The number of symbols output in the current state. Byte-wide implementations are optionally permitted to count symbol pairs, rather than symbols (i.e. odd symbols can be ignored);

d) **PA_max:** The maximum number of preamble symbols that can be received in Hybrid mode before the next cycle is considered unsynchronized. At the input to the Target Smoother, PA_max may be implemented as any value within the range:

$$6 \text{ symbols} \leq PA\_max \leq 10 \text{ symbols}$$

e) **C_Flag:** Indicates that the current SDU is being processed as a cycle in Hybrid mode;

f) **S_Flag:** Indicates that the current or next preamble should be smoothed in Hybrid mode;

g) **T_Flag:** Indicates that the current SDU should not be stripped in Basic mode.

8.8.2.1 State TS0: Preamble (PA)

In this state the Smoother function is processing preamble symbols. The smoother contracts when excess preamble symbols beyond the thresholds are processed. Out_Ct counts the number of output preamble symbols for threshold comparison. For interoperability, the smoother is not required to delete symbols in the exact sequence described by the PA_Actions, nor to delete non-Idle symbols in Basic mode, provided that the counters are accurate upon exit from the PA state.

a) **TS(00): Reset:** When a PHY_Reset signal is received or optionally when PH_Invalid is detected as input, Reset_Actions shall be performed. Reset_Actions initializes the smoother.

b) **TS(01): Start of SDU:** When a Starting Delimiter (JK) symbol pair is detected as input, Start_Actions shall be performed and a transition to State TS1 shall occur. Start_Actions attempts to extend short preambles by inserting Idle symbols, with a corresponding expansion of the smoother. A symbol-wide implementation is permitted to trigger this transition when a J symbol is detected as input; however, the J symbol shall not be transmitted unless it is immediately followed by a K symbol. In Basic mode, an implementation is permitted to use any target threshold value (Target_th) between four and 14 symbols when Start_Actions is invoked

after a token or partial SDU (stripped SDU or format error). On entry to Hybrid mode, the Limit Smoother shall be recentred (and the S_Flag set) either before or after the first Cycle (the state machine shows this action occurring after the first Cycle).

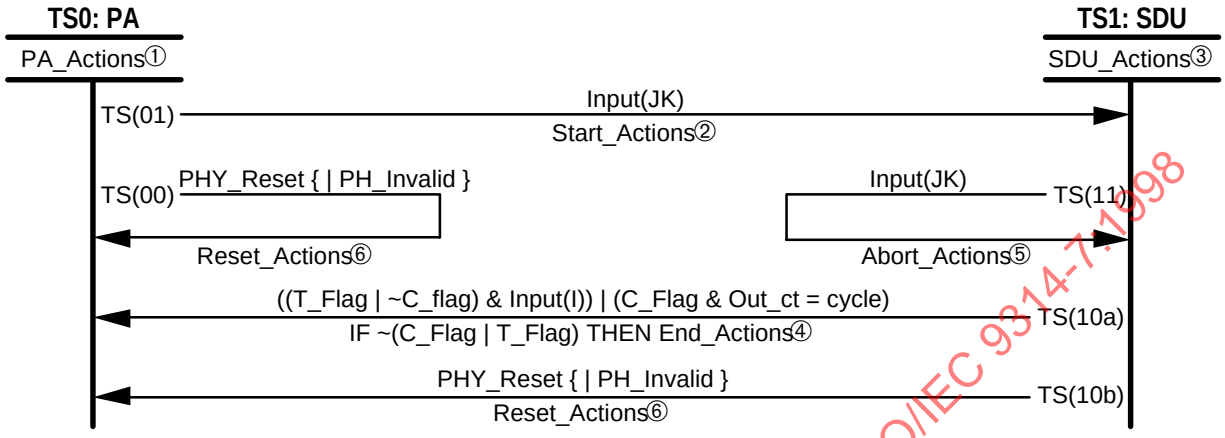


Figure 5 – Target Smoother state diagram (Part 1 of 2)

1. PA_Actions:
   ON entry DO
   CLEAR Out_ct
   FOR EACH Input(symbol) DO
   IF { (C_Flag & Target_ct > Target_cntr & Out_ct = 4) | }
   (Target_ct > 0 & Out_ct ≥ Target_th)
   THEN DEC Target_ct
   ELSE Output(I); INC Out_ct
2. Start_Actions:
   { IF Out_ct > PA_max
   THEN CLEAR S_Flag }
   WHILE { (C_Flag & ~S_Flag & Target_ct < Target_cntr) | }
   (Target_ct < Target_max & Out_ct < Target_th)
   DO INC Target_ct; Output(I); INC Out_ct
   { SET S_Flag = C_Flag }
   { IF C_Flag & output is byte granular « byte granular Target Smoother option »
   THEN IF Out_Ct < 5
   THEN SET Target_th = 6
   ELSE SET Target_th = 4 }
3. SDU_Actions:
   ON entry DO
   CLEAR T_Flag, Out_ct
   FOR EACH Input(symbol) DO
   { IF Out_ct = 3 & ~T_Flag
   THEN IF ~C_Flag & prev = (R | S) & Input(R | S)
   THEN SET C_Flag; SET Target_th = 5
   IF prev = (8 | C) & Input(n)
   THEN CLEAR C_Flag, S_Flag; SET Target_th =14
   { IF prev = T & Input(T)
   THEN SET T_Flag; CLEAR S_Flag }
   SET prev = Input(symbol); }
   IF Input(~n) { & ~C_Flag & (Out_ct > 3 | ~Input(R or S)) }
   THEN SET T_Flag
   Output(Input(symbol)); INC Out_ct
4. End_Actions:
   WHILE Target_Ct > 0 & Out_ct > 0
   DO DEC Target_ct, Out_ct
5. Abort_Actions:
   { CLEAR S_Flag;
   { WHILE C_Flag & Target_ct > Target_cntr & Out_ct > 0
   DO DEC Target_ct, Out_ct
   WHILE C_Flag & Target_ct < Target_cntr
   DO INC Target_ct; Output(I); INC Out_ct
   IF C_Flag
   THEN SET S_Flag; SET Target_th = 5 } }
6. Reset_Actions:
   CLEAR { C_Flag, S_Flag, } Target_ct; SET Target_th =14

Figure 5 – Target Smoother state diagram (Part 2 of 2)

8.8.2.2 State TS1: Service_Data_Unit (SDU)

In this state the Smoother function is processing SDU (frame, token or cycle) symbols. The smoother outputs all input symbols in this state. The C_Flag shall be set on entry to Hybrid mode and cleared on return to Basic mode. The T_Flag shall be set by the first occurrence of a non-data symbol in Basic mode, and may optionally be set when the HRC abort sequence (JKTT) is detected. Out_Ct shall count the number of output SDU symbols, to ensure that End_Actions does not delete symbols prior to the start of a Basic mode SDU, or the end of a Hybrid mode SDU (cycle).

a) **TS(10a): End of SDU:** A transition to State TS0 shall occur when an Idle (I) symbol is detected as input in Basic mode or while the T_Flag is set, or at the end of a cycle in Hybrid mode. If neither the C_Flag nor the T_Flag is set, indicating a stripped Basic mode SDU, then End_Actions shall be performed to reclaim space by deleting previous SDU output symbols and contracting the smoother, or by replacing them with Idles without contracting the smoother. A byte-wide implementation shall trigger this transition in Basic mode when an Idle is detected as the first symbol of a symbol pair, and is optionally permitted to trigger this transition after an Idle is processed as the second symbol of a symbol pair. This transition may optionally occur for other input conditions that terminate the SDU (e.g. in Basic mode, format errors or end of FS field; or end of HRC abort sequence (JKTT) in either Basic or Hybrid mode), provided that End_Actions does not lose completed SDUs or format errors.

b) **TS(10b): Reset:** A transition to State TS0 shall occur when a PHY_Reset signal is received or optionally when PH_Invalid is detected as input, and Reset_Actions shall be performed. Reset_Actions initializes the smoother.

c) **TS(11): Start of SDU:** When a Starting Delimiter (JK) symbol pair is detected as input, the Abort_Actions shall be performed to recentre the Target Smoother if in Hybrid mode (by moving Target_ct toward Target_cntr if C_Flag is set) and a transition to State TS1 shall occur. No preamble adjustment is required on this transition in Basic mode or in implementations that set the T_Flag when the HRC Abort sequence (JKTT) is detected. A symbol-wide implementation is permitted to trigger this transition when a J symbol is detected as input; however, the J symbol shall not be transmitted unless it is immediately followed by a K symbol.

8.9 Repeat Filter function

Certain node configurations require that PHY be able to repeat the PH_UNITDATA.indication symbol stream received on an inbound Physical Link directly as a PH_UNITDATA.request symbol stream on an outbound Physical Link without an intervening Data Link Layer entity (e.g. on the secondary logical ring without a second MAC in the station). Consequently, a Repeat Filter function is required after the PH_UNITDATA.request interface to the outbound Physical Link.

The repeat filter function prevents propagation of code violations and invalid line states from the inbound link to the outbound link, while permitting propagation of lost frames and errors within cycles so they can be correctly counted by the next MAC or HRC entity, respectively, in the logical ring.

Figure 6 shows the Repeat Filter function expressed as a state diagram. In the state diagram, states are shown as vertical staffs and state transitions as horizontal arrows, with the triggering event or condition above the shaft and any action beneath the shaft.

This state machine defines the operation of the Repeat Filter function located immediately before the Encode function. Any implementation that is interoperable with this state machine is permitted unless otherwise prohibited by this part of ISO/IEC 9314.

The state machine uses the following variables:

a) **Out_ct:** The number of symbols output in the current state. Byte-wide implementations are optionally permitted to count symbol pairs, rather than symbols (i.e. odd symbols can be ignored);

b) **C_Flag:** Indicates that the current SDU is being processed as a cycle in Hybrid mode;

c) **T_Flag:** Indicates that the current SDU should not be stripped in Basic mode.

8.9.1 State RF0: IDLE

In this state the Repeat Filter function generates Idle symbols.

a) **RF(00): Reset:** When a PHY_Reset signal is received or optionally when PH_Invalid is detected as input, the optional C_Flag shall be cleared to indicate exit from Hybrid mode. This transition shall take precedence over any other transition if their respective conditions are satisfied simultaneously.

b) **RF(01): Start of SDU:** When a Starting Delimiter (JK) symbol pair is detected as input, a transition to State RF1 shall occur. The T_Flag and Out_ct are cleared. A symbol-wide implementation is permitted to trigger this transition when a J symbol is detected as input; however, the J symbol shall not be transmitted unless it is immediately followed by a K symbol.

8.9.2 State RF1: REPEAT

In this state the Repeat Filter function repeats SDU symbols. The C_Flag shall be set on entry to Hybrid mode and cleared on return to Basic mode. The (optional) T_Flag shall be set when a T symbol is detected as the first symbol of a symbol pair in Basic mode, and may optionally be set when a T symbol is detected as the second symbol of a symbol pair in Basic mode or when the HRC abort sequence (JKTT) is detected. Out_Ct shall count the number of output SDU symbols in Hybrid mode, to properly terminate the SDU (cycle).

a) **RF(10a): End of SDU:** A transition to State RF0 shall occur when an Idle (I) symbol is detected as input in Basic mode or while the T_Flag is set, or at the end of a cycle in Hybrid mode. A byte-wide implementation shall trigger this transition in Basic mode when an Idle is detected as the first symbol of a symbol pair, and is optionally permitted to trigger this transition after an Idle is processed as the second symbol of a symbol pair. This transition may optionally occur for other input conditions that terminate the SDU (e.g. in Basic mode, end of FS field or if the first detected T symbol is the second symbol of a symbol pair, or end of HRC abort sequence (JKTT) in either Basic or Hybrid mode).

b) **RF(10b): Reset:** A transition to State RF0 shall occur when a PHY_Reset signal is received or optionally when PH_Invalid is detected as input, and the optional C_Flag shall be cleared to indicate exit from Hybrid mode. This transition shall take precedence over any other transition if their respective conditions are satisfied simultaneously.

c) **RF(11): Start of SDU:** When a Starting Delimiter (JK) symbol pair is detected as input, a transition to State RF1 shall occur. The T_Flag and Out_ct are cleared. A symbol-wide implementation is permitted to trigger this transition when a J symbol is detected as input; however, the J symbol shall not be transmitted unless it is immediately followed by a K symbol. This transition shall take precedence over transition RF(10a) if their respective conditions are satisfied simultaneously.

d) **RF(12): Filter SDU:** When a potentially bad symbol is detected as input, a transition to State RF2 shall occur. This transition may optionally occur when an R or S symbol is detected in Basic mode if the T_Flag is not set.

8.9.3 State RF2: FILTER

In this state the Repeat Filter function filters SDU symbols. In Basic mode an H symbol shall be generated unless the T_Flag is set; otherwise a T symbol shall be generated. In Hybrid mode, an L symbol shall be generated unless it would create a false Embedded Delimiter pair (IL); otherwise an I symbol shall be generated. Out_Ct shall count the number of output SDU symbols, to properly terminate the SDU (cycle, frame or fragment).

a) **RF(20a): End of SDU:** A transition to State RF0 shall occur when an Idle (I) symbol is detected as input in Basic mode or while the T_Flag is set, or after four filter symbols have been generated in Basic mode, or at the end of a cycle in Hybrid mode. This transition may optionally occur after one or more filter (T) symbols have been generated if the T_Flag is set. A byte-wide implementation shall trigger this transition in Basic mode when an Idle is detected as the first symbol of a symbol pair, or after two pairs of filter symbols have been generated; and is optionally permitted to trigger this transition in Basic mode after an Idle is processed as the second symbol of a symbol pair, or after four filter symbols have been generated. This transition shall take precedence over transition RF(21b) if their respective conditions are satisfied simultaneously.

b) **RF(20b): Reset:** A transition to State RF0 shall occur when a PHY_Reset signal is received or optionally when PH_Invalid is detected as input, and the optional C_Flag shall be cleared to indicate exit from Hybrid mode. This transition shall take precedence over any other transition if their respective conditions are satisfied simultaneously.

c) **RF(21a): Start of SDU:** When a Starting Delimiter (JK) symbol pair is detected as input, a transition to State RF1 shall occur. The T_Flag and Out_ct are cleared. A symbol-wide implementation is permitted to trigger this transition when a J symbol is detected as input; however, the J symbol shall not be transmitted unless it is immediately followed by a K symbol. This transition shall take precedence over transition RF(20a) if their respective conditions are satisfied simultaneously.

d) **RF(21b): Resume SDU:** When a good symbol is detected as input, a transition to State RF1 shall occur in Hybrid mode, and may optionally occur in Basic mode if the T_Flag is set.