the PMA performs a conversion from NRZI format and generates a PMA_UNITDATA.indicate (rx_code-bit) primitive with the same logical value for the rx_code-bit parameter.

### 24.3.4.3 Carrier detect

The PMA Carrier Detect process provides repeater clients an indication that a carrier event has been sensed and an indication if it is deemed in error. A carrier event is defined as receipt of two non-contiguous ZEROS within any 10 rx_code-bits. A carrier event is in error if it does not start with an SSD. The Carrier Detect process performs this function by continuously monitoring the code-bits being delivered by the RX process, and checks for specific patterns which indicate non-IDLE activity and SSD bit patterns.

The Carrier Detect process collects code-bits from the PMD RX process. r_bits [9:0] represents a sliding, 10-bit window on the code-bit sequence, with newly received code-bits from the RX process being shifted into r_bits [0]. The process shifts the r_bits vector to the left, inserts the newly received code-bit into position 0, and waits for the next PMD.UNITDATA.indicate before repeating the operation. This is depicted in figure 24-13. The Carrier Detect process monitors the r_bits vector until it detects two noncontiguous ZEROS in the incoming code-bit sequence. This signals a transition of carrier_status from OFF to ON. Each new carrier is further examined for a leading SSD (1100010001) with rxerror_status set to ERROR if it is not confirmed. A pattern of 10 contiguous ONEs in the stream indicates a return to carrier_status = OFF. Code-bit patterns of contiguous ONEs correspond to IDLE code-groups in the PCS, per the encoding specified in 24.2.2.1.

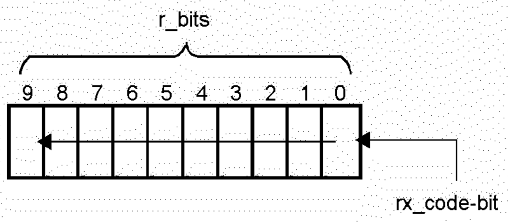


**Figure 24-13—Carrier Detect reference diagram**

The PMA shall, if it is supporting a repeater, implement the Carrier Detect process as depicted in figure 24-14 including compliance with the associated state variables as specified in 24.3.3.

### 24.3.4.4 Link Monitor

The Link Monitor process is responsible for determining whether the underlying receive channel is providing reliable data. Failure of the underlying channel typically causes the PMA's client to suspend normal actions. The Link Monitor process takes advantage of the PMD sublayer's continuously signaled transmission scheme, which provides the PMA with a continuous indication of signal detection on the channel through signal_status as communicated by the PMD_SIGNAL.indicate primitive. It responds to control by Auto-Negotiation, when implemented, which is effected through the link_control parameter of PMA_SIGNAL request.

The Link Monitor process monitors signal_status, setting link_status to FAIL whenever signal_status is OFF or when Auto-Negotiation sets link_control to DISABLE. The link is deemed to be reliably operating when signal_status has been continuously ON for a period of time. This period is implementation dependent but not less than 330 μs or greater than 1000 μs. If so qualified, Link Monitor sets link_status to READY in order to synchronize with Auto-Negotiation, when implemented. Auto-Negotiation permits full operation by setting link_control to ENABLE. When Auto-Negotiation is not implemented, Link Monitor operates with link_control always set to ENABLE.

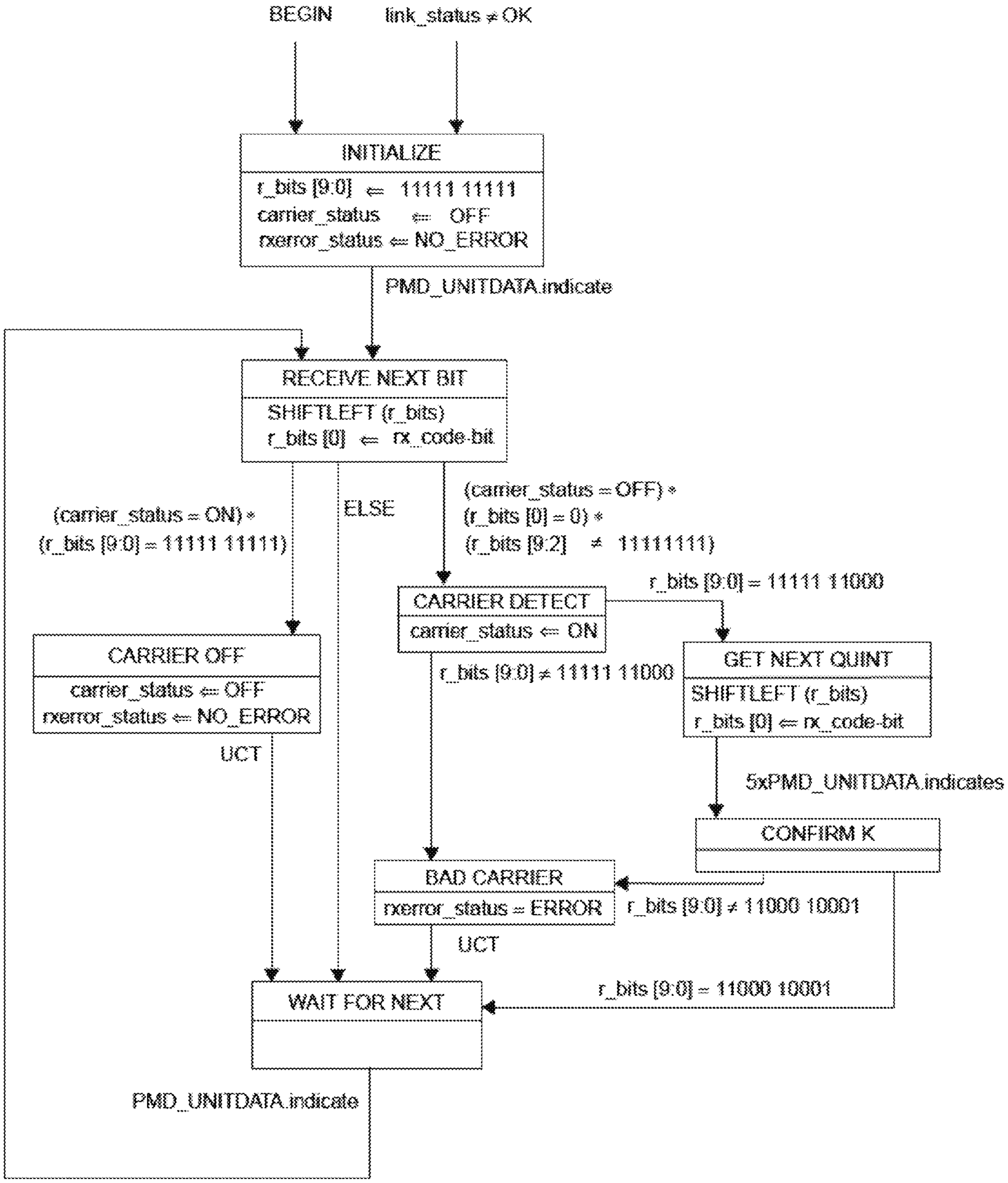
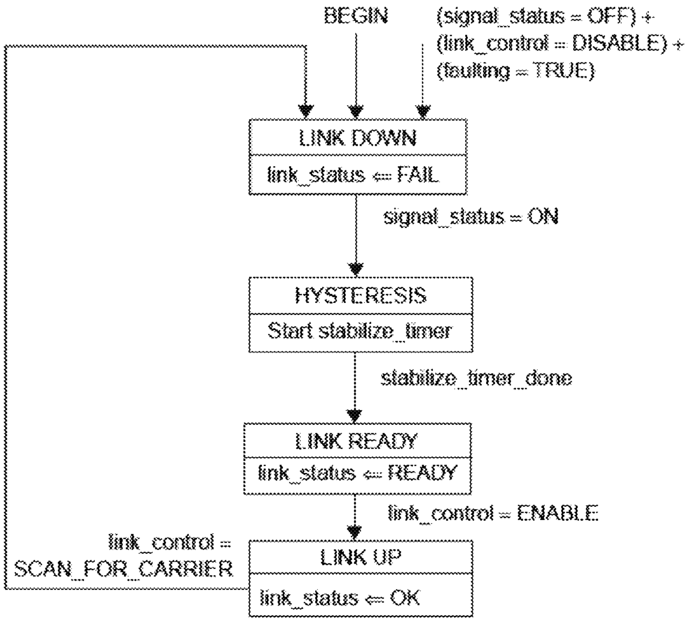


Figure 24-14—Carrier Detect state diagram

The PMA shall implement the Link Monitor process as depicted in figure 24-15 including compliance with the associated state variables as specified in 24.3.3.

### 24.3.4.5 Far-End Fault Generate

Far-End Fault Generate simply passes tx_code-bits to the TX process when signal_status=ON. When signal_status=OFF, it repetitively generates each cycle of the Far-End Fault Indication until signal_status is reasserted.



NOTE—The variables link_control and link_status are designated as link_control_[TX] and link_status_[TX], respectively, by the Auto-Negotiation Arbitration state diagram (figure 28-16).

**Figure 24-15—Link Monitor state diagram**

If Far-End Fault is implemented, the PMA shall implement the Far-End Fault Generate process as depicted in figure 24-16 including compliance with the associated state variables as specified in 24.3.3.

### 24.3.4.6 Far-End Fault Detect

Far-End Fault Detect passively monitors the rx_code-bit stream from the RX process for the Far-End Fault Indication. It does so by maintaining counters for the number of consecutive ONEs seen since the last ZERO (num_ones) and the number of cycles of 84 ONEs and a single ZERO (num_cycles). The Far-End Fault Indication is denoted by three or more cycles, each of 84 ONEs and a single ZERO. Note that the number of consecutive ONEs may exceed 84 on the first cycle.

If Far-End Fault is implemented, the PMA shall implement the Far-End Fault Detect process as depicted in figure 24-17 including compliance with the associated state variables as specified in 24.3.3.

## 24.4 Physical Medium Dependent (PMD) sublayer service interface

### 24.4.1 PMD service interface

The following specifies the services provided by the PMD. The PMD is a sublayer within 100BASE-X and may not be present in other 100BASE-T PHY specifications. PMD services are described in an abstract manner and do not imply any particular implementation. It should be noted that these services are functionally identical to those defined in the FDDI standards, such as ISO 9314-3: 1990 and ANSI X3.263: 199X, with two exceptions:

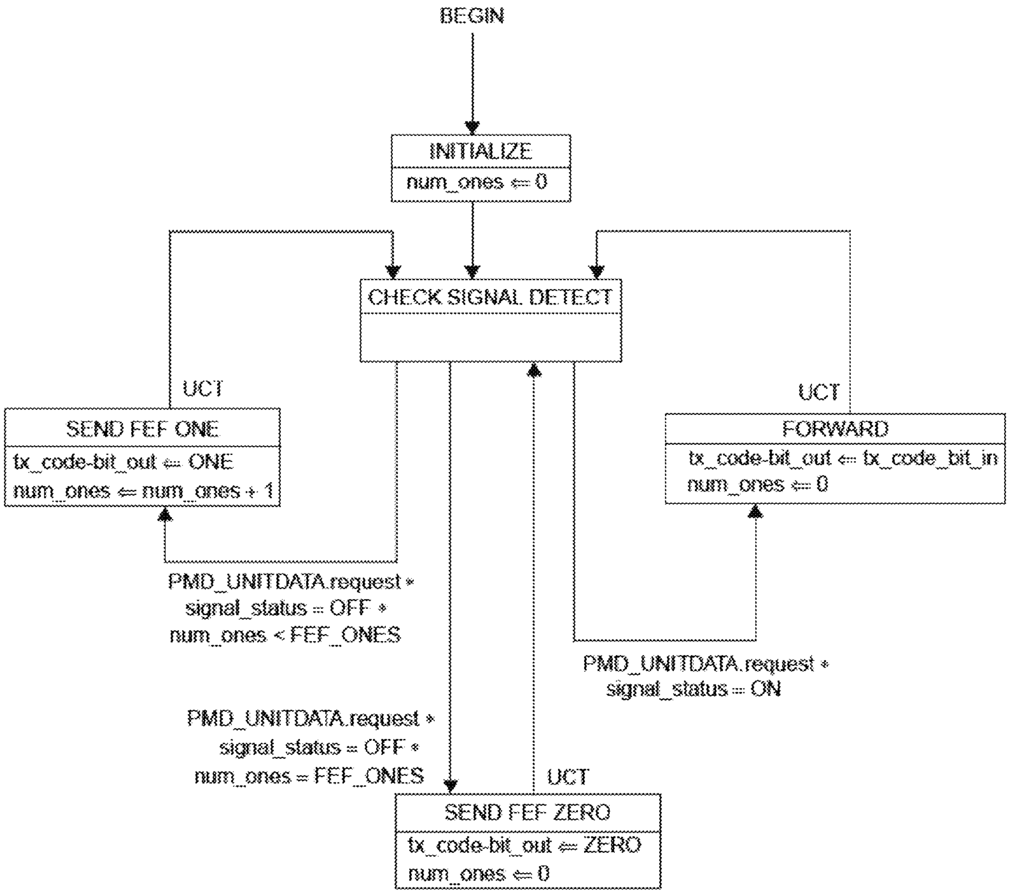


**Figure 24-16—Far-End Fault Generate state diagram**

a) 100BASE-X does not include a Station Management (SMT) function; therefore the PMD-to-SMT interface defined in ISO 9314-3: 1990 and ANSI X3.263: 199X.
b) 100BASE-X does not support multiple instances of a PMD in service to a single PMA; therefore, no qualifiers are needed to identify the unique PMD being referenced.

There are also *editorial* differences between the interfaces specified here and in the referenced standards, as required by the context of 100BASE-X.

The PMD Service Interface supports the exchange of nrzi-bits between PMA entities. The PMD translates the nrzi-bits to and from signals suitable for the specified medium.

The following primitives are defined:

PMD_UNITDATA.request
PMD_UNITDATA.indicate
PMD_SIGNAL.indicate

### 24.4.1.1 PMD_UNITDATA.request

This primitive defines the transfer of data (in the form of nrzi-bits) from the PMA to the PMD.

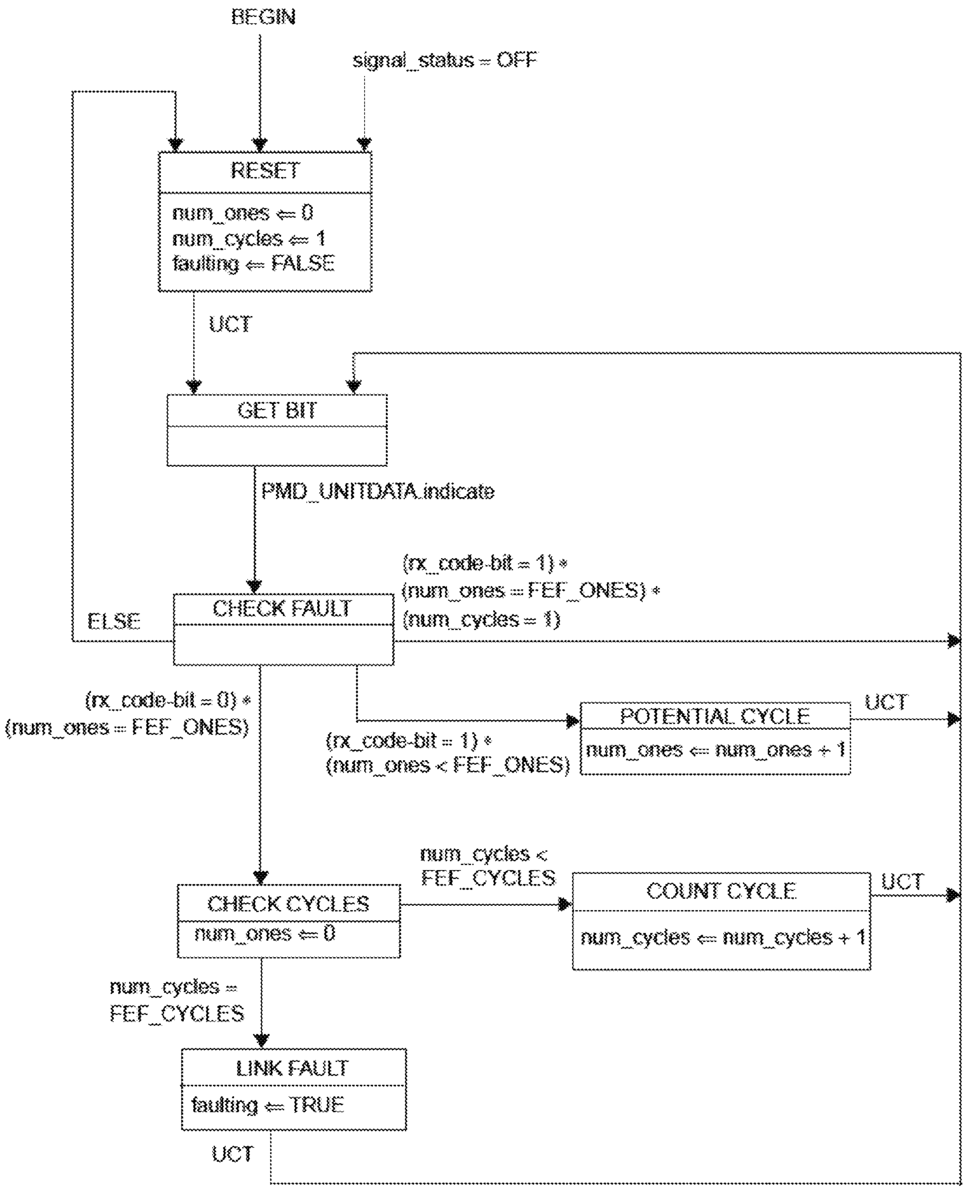


**Figure 24-17—Far-End Fault Detect state diagram**

### 24.4.1.1.1 Semantics of the service primitive

PMD_UNITDATA.request (tx_nrzi-bit)

The data conveyed by PMD_UNITDATA.request is a continuous sequence of nrzi-bits. The tx_nrzi-bit parameter can take one of two values: ONE or ZERO.

### 24.4.1.1.2 When generated

The PMA continuously sends, at a nominal 125 Mb/s rate, the PMD the appropriate nrzi-bits for transmission on the medium.

#### 24.4.1.1.3 Effect of receipt

Upon receipt of this primitive, the PMD converts the specified nrzi-bit into the appropriate signals on the MDI.

#### 24.4.1.2 PMD_UNITDATA.indicate

This primitive defines the transfer of data (in the form of nrzi-bits) from the PMD to the PMA.

#### 24.4.1.2.1 Semantics of the service primitive

PMD_UNITDATA.indicate (rx_nrzi-bit)

The data conveyed by PMD_UNITDATA.indicate is a continuous nrzi-bit sequence. The rx_nrzi-bit parameter can take one of two values: ONE or ZERO.

#### 24.4.1.2.2 When generated

The PMD continuously sends nrzi-bits to the PMA corresponding to the signals received from the MDI.

#### 24.4.1.2.3 Effect of receipt

The effect of receipt of this primitive by the client is unspecified by the PMD sublayer.

#### 24.4.1.3 PMD_SIGNAL.indicate

This primitive is generated by the PMD to indicate the status of the signal being received from the MDI.

#### 24.4.1.3.1 Semantics of the service primitive

PMD_SIGNAL.indicate (signal_status)

The signal_status parameter can take on one of two values: ON or OFF, indicating whether the quality and level of the received signal is satisfactory (ON) or unsatisfactory (OFF). When signal_status = OFF, then rx_nrzi-bit is undefined, but consequent actions based on PMD_SIGNAL.indicate, where necessary, interpret rx_nrzi-bit as logic ZERO.

#### 24.4.1.3.2 When generated

The PMD generates this primitive to indicate a change in the value of signal_status.

#### 24.4.1.3.3 Effect of receipt

The effect of receipt of this primitive by the client is unspecified by the PMD sublayer.

### 24.4.2 Medium Dependent Interface (MDI)

The MDI, a physical interface associated with a PMD, is comprised of an electrical or optical medium connector. The 100BASE-X MDIs, defined in subsequent clauses, are specified by reference to the appropriate FDDI PMD, such as in ISO 9314-3: 1990 and ANSI X3.263: 199X, together with minor modifications (such as connectors and pin-outs) necessary for 100BASE-X.

## 24.5 Compatibility considerations

There is no requirement for a compliant device to implement or expose any of the interfaces specified for the PCS, PMA, or PMD. However, if an exposed interface is provided to the PCS, it shall comply with the requirements for the MII, as specified in clause 22.

## 24.6 Delay constraints

Proper operation of a CSMA/CD LAN demands that there be an upper bound on the propagation delays through the network. This implies that MAC, PHY, and repeater implementors must conform to certain delay minima and maxima, and that network planners and administrators conform to constraints regarding the cable topology and concatenation of devices. MAC constraints are contained in clause 21. Topological constraints are contained in clause 29.

The reference point for all MDI measurements is the 50% point of the mid-cell transition corresponding to the reference code-bit, as measured at the MDI. Although 100BASE-TX output is scrambled, it is assumed that these measurements are made via apparatuses that appropriately account for this.

### 24.6.1 PHY delay constraints (exposed MII)

Every 100BASE-X PHY with an exposed MII shall comply with the bit delay constraints specified in table 24-2. These figures apply for all 100BASE-X PMDs.

**Table 24-2—MDI to MII delay constraints (exposed MII)**

| Sublayer measurement points | Event | Min (bits) | Max (bits) | Input timing reference | Output timing reference |
|---|---|---|---|---|---|
| MII ⇔ MDI | TX_EN Sampled to MDI Output | 6 | 14 | TX_CLK rising | 1st bit of /J/ |
| | MDI input to CRS assert | | 20 | 1st bit of /J/ | |
| | MDI input to CRS de-assert (aligned) | 13 | 24 | 1st bit of /T/ | |
| | MDI input to CRS de-assert (unaligned) | 13 | 24 | 1st ONE | |
| | MDI input to COL assert | | 20 | 1st bit of /J/ | |
| | MDI input to COL de-assert (aligned) | 13 | 24 | 1st bit of /T/ | |
| | MDI input to COL de-assert (unaligned) | 13 | 24 | 1st ONE | |
| | TX_EN sampled to CRS assert | 0 | 4 | TX_CLK rising | |
| | TX_EN sampled to CRSde-assert | 0 | 16 | TX_CLK rising | |

### 24.6.2 DTE delay constraints (unexposed MII)

Every 100BASE-X DTE with no exposed MII shall comply with the bit delay constraints specified in table 24-3. These figures apply for all 100BASE-X PMDs.

### 24.6.3 Carrier de-assertion/assertion constraint

To ensure fair access to the network, each DTE shall, additionally, satisfy the following:

Table 24-3—DTE delay constraints (unexposed MII)

| Sublayer measurement points | Event | Min (bits) | Max (bits) | Input timing reference | Output timing reference |
|---|---|---|---|---|---|
| MAC ⇔ MDI | MAC transmit start to MDI output | | 18 | | 1st bit of /J/ |
| | MDI input to MDI output (worst-case nondeferred transmit) | | 54 | 1st bit of /J/ | 1st bit of /J/ |
| | MDI input to collision detect | | 28 | 1st bit of /J/ | |
| | MDI input to MDI output = Jam (worst case collision response) | | 54 | 1st bit of /J/ | 1st bit of jam |

(MAX MDI to MAC Carrier De-assert Detect) – (MIN MDI to MAC Carrier Assert Detect) < 13

## 24.7 Environmental specifications

All equipment subject to this clause shall conform to the requirements of 14.7 and applicable sections of ISO/IEC 11801: 1995.

## 24.8 Protocol Implementation Conformance Statement (PICS) proforma for clause 24, Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 100BASE-X[23]

### 24.8.1 Introduction

The supplier of a protocol implementation that is claimed to conform to IEEE Std 802.3u-1995, Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 100BASE-X, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in clause 21.

### 24.8.2 Identification

#### 24.8.2.1 Implementation identification

| | |
|---|---|
| Supplier | |
| Contact point for enquiries about the PICS | |
| Implementation Name(s) and Version(s) | |
| Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s) | |
| NOTES<br>1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.<br>2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model). | |

#### 24.8.2.2 Protocol summary

| | |
|---|---|
| Identification of protocol standard | IEEE Std 802.3u-1995, Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 100BASE-X |
| Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS | |
| Have any Exception items been required? No [ ] Yes [ ]<br>(See clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3u-1995.) | |
| Date of Statement | |

---

[23] *Copyright release for PICS proformas* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

### 24.8.2.3 Major capabilities/options

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| *DTE | Supports DTE without MII | 24.4 | O/1 | | |
| *REP | Supports Repeater without MII | 24.4 | O/1 | | |
| *MII | Supports exposed MII interface | 24.4 | O/1 | | |
| *PCS | Implements PCS functions | 24.2 | REP: O<br>DTE: M<br>MII: M | | |
| PMA | Implements PMA RX, TX and Link Monitor functions | 24.3 | M | | |
| *NWC | Medium capable of supporting Auto-Negotiation | | O | | See clause 28 |
| *FEF | Implements Far-End Fault | 24.3.2.1 | NWC: X | | |
| NWY | Supports Auto-Negotiation (clause 28) | | NWC: O | | See clause 28 |

## 24.8.3 PICS proforma tables for the Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer, type 100BASE-X

### 24.8.3.1 General compatibility considerations

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| GN1 | Compliance with MII requirements | 24.4 | MII:M | | See clause 22 |
| GN2 | Environmental specifications | 24.7 | M | | |

### 24.8.3.2 PCS functions

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| PS1 | Transmit Bits process | 24.2.3 | PCS:M | | |
| PS2 | Transmit process | 24.2.4.2 | PCS:M | | |
| PS3 | Receive Bits process | 24.2.4.3 | PCS:M | | |
| PS4 | Receive process | 24.2.4.4 | PCS:M | | |
| PS5 | Carrier Sense process | 24.2.4.5 | PCS:M | | |

### 24.8.3.3 PMA functions

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| PA1 | TX process | 24.3.4.1 | M | | |
| PA2 | RX process | 24.3.4.2 | M | | |
| PA3 | Carrier Detect process | 24.3.2.1 | REP: M | | |
| PA4 | Link Monitor process | 24.3.4.4 | M | | |
| PA5 | Far-End Fault Generate process | 24.3.4.5 | FEF: M | | |
| PA6 | Far-End Fault Detect process | 24.3.4.6 | FEF: M | | |

### 24.8.3.4 Timing

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| TM1 | Support for MII signals TX_CLK and RX_CLK | 24.2.2.3 | MII:M | | See clause 22 |
| TM2 | Accuracy of code-bit_timer | 24.2.3 | M | | |
| TM3 | Compliance with PHY bit delay constraints | 24.6.1 | MII:M<br>REP: O | | |
| TM4 | Compliance with DTE bit delay constraints | 24.6.2 | DTE:M | | |
| TM5 | Compliance with Carrier Deassert/Assert Constraint | 24.6.3 | DTE:M | | |

# 25. Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-TX

## 25.1 Overview

This clause specifies the 100BASE-X PMD (including MDI) and baseband medium for twisted-pair wiring, 100BASE-TX. In order to form a complete 100BASE-TX Physical Layer it shall be integrated with the 100BASE-X PCS and PMA of clause 24, which are assumed incorporated by reference. As such, the 100BASE-TX PMD shall comply with the PMD service interface specified in 24.4.1.

## 25.2 Functional specifications

The 100BASE-TX PMD (and MDI) is specified by incorporating the FDDI TP-PMD standard, ANSI X3.263: 199X (TP-PMD), by reference, with the modifications noted below. This standard provides support for Category 5 unshielded twisted pair (UTP) and shielded twisted pair (STP). For improved legibility in this clause, ANSI X3.263: 199X (TP-PMD), will henceforth be referred to as TP-PMD.

## 25.3 General exceptions

The 100BASE-TX PMD is precisely the PMD specified as TP-PMD, with the following general modifications:

a) The Scope and General description discussed in TP-PMD 1 and 5 relate to the use of those standards with an FDDI PHY, ISO 9314-1: 1989, and MAC, ISO 9314-2: 1989. These sections are not relevant to the use of the PMD with 100BASE-X.
b) The Normative references, Definitions and Conventions of TP-PMD 2, 3, and 4 are used only as necessary to interpret the applicable sections of TP-PMD referenced in this clause.
c) The PMD Service Specifications of TP-PMD 6 are replaced by those specified in 24.4.1. The 100BASE-TX PMD Service specification is a proper subset of the PMD Service Specification in TP-PMD.
d) There are minor terminology differences between this standard and TP-PMD that do not cause ambiguity. The terminology used in 100BASE-X was chosen to be consistent with other IEEE 802 standards, rather than with FDDI. Terminology is both defined and consistent within each standard. Special note should be made of the interpretations shown in table 25-1.

**Table 25-1—Interpretation of general FDDI terms and concepts**

| FDDI term or concept | Interpretation for 100BASE-TX |
|---|---|
| bypass | <unused> |
| Connection Management (CMT) | <no comparable entity> |
| frame | stream |
| Halt Line State (HLS) | <unused> |
| hybrid mode | <no comparable entity> |
| MAC (or MAC-2) | MAC |
| Master Line State (MLS) | <unused> |
| maximum frame size = 9000 symbols | maximum stream size = 3054 code-groups |
| PHY (or PHY-2) | PMA; i.e., PMD client |

Table 25-1—Interpretation of general FDDI terms and concepts *(Continued)*

| FDDI term or concept | Interpretation for 100BASE-TX |
|---|---|
| PHY Service Data Unit (SDU) | stream |
| PM_SIGNAL.indication (Signal_Detect) | PMD_SIGNAL.indicate (signal_status) |
| PM_UNITDATA.indication (PM_Indication) | PMD_UNITDATA.indicate (nrzi-bit) |
| PM_UNITDATA request (PM_Request) | PMD_UNITDATA request (nrzi-bit) |
| preamble | inter-packet IDLEs |
| Quiet Line State (QLS) | <unused> |
| SM_PM_BYPASS request (Control_Action) | Assume:<br>SM_PM_BYPASS request(Control_Action = Insert) |
| SM_PM_CONTROL request (Control_Action) | Assume:<br>SM_PM_CONTROL request (Control_Action = Transmit_Enable) |
| SM_PM_SIGNAL.indication (Signal_Detect) | <unused> |
| Station Management (SMT) | <no comparable entity> |
| symbol | code-group |

## 25.4 Specific requirements and exceptions

The 100BASE-TX PMD (including MDI) and baseband medium shall comply to the requirements of TP-PMD, 7, 8, 9, 10, and 11, and normative annex A with the exceptions listed below. In TP-PMD, informative annexes B, C, E, F, G, I, and J, with exceptions listed below, provide additional information useful to PMD sublayer implementors. Where there is conflict between specification in TP-PMD and those in this standard, those of this standard shall prevail.

### 25.4.1 Change to 7.2.3.1.1, "Line state patterns"

Descrambler synchronization on the Quiet Line State (QLS), Halt Line State (HLS), and Master Line State (MLS) Line State Patterns cited in TP-PMD 7.2.3.1.1 is optional.

### 25.4.2 Change to 7.2.3.3, "Loss of synchronization"

The synchronization error triggered by PH_Invalid as defined in TP-PMD 7.2.3.3a is not applicable.

### 25.4.3 Change to table 8-1, "Contact assignments for unshielded twisted pair"

100BASE-TX for unshielded twisted pair adopts the contact assignments of 10BASE-T. Therefore, the contact assignments shown in TP-PMD table 8-1 shall instead be as depicted in table 25-2.

### 25.4.4 Deletion of 8.3, "Station labelling"

Clause 8.3 of TP-PMD shall not be applied to 100BASE-TX.

### 25.4.5 Change to 9.1.9, "Jitter"

The jitter measurement specified in 9.1.9 of TP-PMD may be performed using scrambled IDLEs.

### Table 25-2—UTP MDI contact assignments

| CONTACT | PHY without internal crossover MDI SIGNAL | PHY with internal crossover MDI SIGNAL |
|---|---|---|
| 1 | Transmit + | Receive + |
| 2 | Transmit – | Receive – |
| 3 | Receive + | Transmit + |
| 4 | | |
| 5 | | |
| 6 | Receive – | Transmit – |
| 7 | | |
| 8 | | |

### 25.4.6 Replacement of 11.2, "Crossover function"

Clause 11.2 of TP-PMD is replaced with the following:

A crossover function compliant with 14.5.2 shall be implemented except that a) the signal names are those used in TP-PMD, and b) the contact assignments for STP are those shown in table 8-2 of TP-PMD. Note that compliance with 14.5.2 implies a recommendation that crossover (for both UTP and STP) be performed within repeater PHYs.

### 25.4.7 Change to A.2, "DDJ test pattern for baseline wander measurements"

The length of the test pattern specified in TP-PMD annex A.2 may be shortened to accommodate feasible 100BASE-X measurements, but shall not be shorter than 3000 code-groups.

NOTE—This pattern is to be applied to the MII. (When applied to the MAC, the nibbles within each byte are to be swapped. E.g., as delivered to the MAC, the test pattern would start, "60 c9 16 ...".)

### 25.4.8 Change to annex G, "Stream cipher scrambling function"

An example of a stream cipher scrambling implementation is shown in TP-PMD annex G. This may be modified to allow synchronization solely on the IDLE sequences between packets.

### 25.4.9 Change to annex I, "Common mode cable termination"

The contact assignments shown in TP-PMD figures I-1 and I-2 shall instead comply with those specified in table 25-2.

## 25.5 Protocol Implementation Conformance Statement (PICS) proforma for clause 25, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-TX[24]

### 25.5.1 Introduction

The supplier of a protocol implementation that is claimed to conform to IEEE Std 802.3u-1995, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-TX, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in clause 21.

### 25.5.2 Identification

#### 25.5.2.1 Implementation identification

| | |
|---|---|
| Supplier | |
| Contact point for enquiries about the PICS | |
| Implementation Name(s) and Version(s) | |
| Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s) | |
| NOTES<br>1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.<br>2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model). | |

#### 25.5.2.2 Protocol summary

| | |
|---|---|
| Identification of protocol standard | IEEE Std 802.3u-1995, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-TX |
| Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS | |
| Have any Exception items been required? No [ ] Yes [ ]<br>(See clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3u-1995.) | |
| Date of Statement | |

---

[24] *Copyright release for PICS proformas* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

### 25.5.3 Major capabilities/options

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| *TXU | Supports unshielded twisted pair | 25.2 | O/1 | | |
| TXS | Supports shielded twisted pair | 25.2 | O/1 | | |

### 25.5.4 PICS proforma tables for the Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-TX

#### 25.5.4.1 General compatibility considerations

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| GN1 | Integrates 100BASE-X PMA and PCS | 25.1 | M | | See clause 24 |

#### 25.5.4.2 PMD compliance

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| PD1 | Compliance with 100BASE-X PMD Service Interface | 25.1 | M | | See 24.2.3 |
| PD2 | Compliance with ANSI X3.237: 199X, 7, 8 (excluding 8.3), 9, 10, 11 and normative annex A, with listed exceptions | 25.4<br>25.4.5 | M | | |
| PD3 | Precedence over ANSI X3.237-199X | 25.4 | M | | |
| PD4 | MDI contact assignments for unshielded twisted pair | 25.4.4<br>25.4.10 | TXU: M | | |
| PD5 | Compliance with crossover function of 14.5.2 with listed adaptations | 25.4.7 | M | | |
| PD6 | Minimum jitter test pattern length | 25.4.8 | M | | 3000 code-groups |

# 26. Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-FX

## 26.1 Overview

This clause specifies the 100BASE-X PMD (including MDI) and fiber optic medium for multi-mode fiber, 100BASE-FX. In order to form a complete 100BASE-FX Physical Layer it shall be integrated with the 100BASE-X PCS and PMA of clause 24, which are assumed incorporated by reference. As such, the 100BASE-FX PMD shall comply with the PMD service interface specified in 24.4.1.

## 26.2 Functional specifications

The 100BASE-FX PMD (and MDI) is specified by incorporating the FDDI PMD standard, ISO 9314-3: 1990, by reference, with the modifications noted below. This standard provides support for two optical fibers. For improved legibility in this clause, ISO 9314-3: 1990 will henceforth be referred to as fiber-PMD.

## 26.3 General exceptions

The 100BASE-FX PMD is precisely the PMD specified as fiber-PMD, with the following general modifications:

a) The Scope and General description discussed in fiber-PMD 1 and 5 relate to the use of those standards with an FDDI PHY, ISO 9314-1: 1989, and MAC, ISO 9314-2: 1989. These clauses are not relevant to the use of the PMD with 100BASE-X.
b) The Normative references, Definitions and Conventions of fiber-PMD 2, 3, and 4 are used only as necessary to interpret the applicable sections of fiber-PMD referenced in this clause.
c) The PMD Service Specifications of fiber-PMD 6 are replaced by those specified in 24.4.1. The 100BASE-FX PMD Service specification is a proper subset of the PMD service specification in fiber-PMD.
d) There are minor terminology differences between this standard and fiber-PMD that do not cause ambiguity. The terminology used in 100BASE-X was chosen to be consistent with other IEEE 802 standards, rather than with FDDI. Terminology is both defined and consistent within each standard. Special note should be made of the interpretations shown in table 26-1.

**Table 26-1—Interpretation of general FDDI terms and concepts**

| FDDI term or concept | Interpretation for 100BASE-X |
|---|---|
| bypass | <unused> |
| Connection Management (CMT) | <no comparable entity> |
| frame | stream |
| Halt Line State (HLS) | <unused> |
| hybrid mode | <no comparable entity> |
| MAC (or MAC-2) | MAC |
| Master Line State (MLS) | <unused> |
| maximum frame size = 9000 symbols | maximum stream size = 3054 code-groups |

**Table 26-1—Interpretation of general FDDI terms and concepts *(Continued)***

| FDDI term or concept | Interpretation for 100BASE-X |
|---|---|
| PHY (or PHY-2) | PMA; i.e., PMD client |
| PHY Service Data Unit (SDU) | stream |
| PM_SIGNAL.indication (Signal_Detect) | PMD_SIGNAL.indicate (signal_status) |
| PM_UNITDATA.indication (PM_Indication) | PMD_UNITDATA.indicate (nrzi-bit) |
| PM_UNITDATA request (PM_Request) | PMD_UNITDATA request (nrzi-bit) |
| preamble | inter-packet IDLEs |
| Quiet Line State (QLS) | <unused> |
| SM_PM_BYPASS request (Control_Action) | Assume:<br>SM_PM_BYPASS request (Control_Action = Insert) |
| SM_PM_CONTROL request (Control_Action) | Assume:<br>SM_PM_CONTROL request (Control_Action = Transmit_Enable) |
| SM_PM_SIGNAL.indication (Signal_Detect) | <unused> |
| Station Management (SMT) | <no comparable entity> |
| symbol | code-group |

## 26.4 Specific requirements and exceptions

The 100BASE-FX PMD (including MDI) and baseband medium shall conform to the requirements of fiber-PMD 8, 9, and 10. In fiber-PMD, informative annexes A through G provide additional information useful to PMD sublayer implementors. Where there is conflict between specifications in fiber-PMD and those in this standard, those of this standard shall prevail.

### 26.4.1 Medium Dependent Interface (MDI)

The 100BASE-FX medium dependent interface (MDI) shall conform to one of the following connectors. The recommended alternative is the Low Cost Fibre Optical Interface Connector.

a) Low Cost Fibre Optical Interface Connector (commonly called the duplex SC connector) as specified in ANSI X3.237-199X, 7.1.1 through 7.3.1, inclusive.
b) Media Interface Connector (MIC) as specified in fiber-PMD 7 and annex F. When the MIC is used, the receptacle shall be keyed as “M”.
c) Optical Medium Connector Plug and Socket (commonly called ST connector) as specified in 15.3.2.

### 26.4.2 Crossover function

A crossover function shall be implemented in every cable-pair link. The crossover function connects the transmitter of one PHY to the receiver of the PHY at the other end of the cable-pair link. For 100BASE-FX, the crossover function is realized in the cable plant.

## 26.5 Protocol Implementation Conformance Statement (PICS) proforma for clause 26, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-FX[25]

### 26.5.1 Introduction

The supplier of a protocol implementation that is claimed to conform to IEEE Std 802.3u-1995, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-FX, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in clause 21.

### 26.5.2 Identification

#### 26.5.2.1 Implementation identification

| | |
|---|---|
| Supplier | |
| Contact point for enquiries about the PICS | |
| Implementation Name(s) and Version(s) | |
| Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s) | |
| NOTES<br>1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.<br>2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model). | |

### 26.5.3 Protocol summary

| | |
|---|---|
| Identification of protocol standard | IEEE Std 802.3u-1995, Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-FX |
| Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS | |
| Have any Exception items been required? No [ ] Yes [ ]<br>(See clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3u-1995.) | |
| Date of Statement | |

---

[25] *Copyright release for PICS proformas* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

### 26.5.4 Major capabilities/options

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| FSC | Supports Low Cost Fibre Optical Interface Connector (duplex SC) | 26.4.2 | O/1 | | Recommended.<br>See ANSI X3.237-199X, 7.1.1 through 7.3.1 |
| *FMC | Supports Media Interface Connector (MIC) | 26.4.2 | O/1 | | See ISO 9314-3: 1990, 7 and annex F |
| FST | Supports Optical Medium Connector Plug and Socket (ST) | 26.4.2 | O/1 | | See 15.3.2 |

### 26.5.5 PICS proforma tables for Physical Medium Dependent (PMD) sublayer and baseband medium, type 100BASE-FX

#### 26.5.5.1 General compatibility considerations

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| GN1 | Integrates 100BASE-X PMA and PCS | 26.1 | M | | See clause 24 |

#### 26.5.5.2 PMD compliance

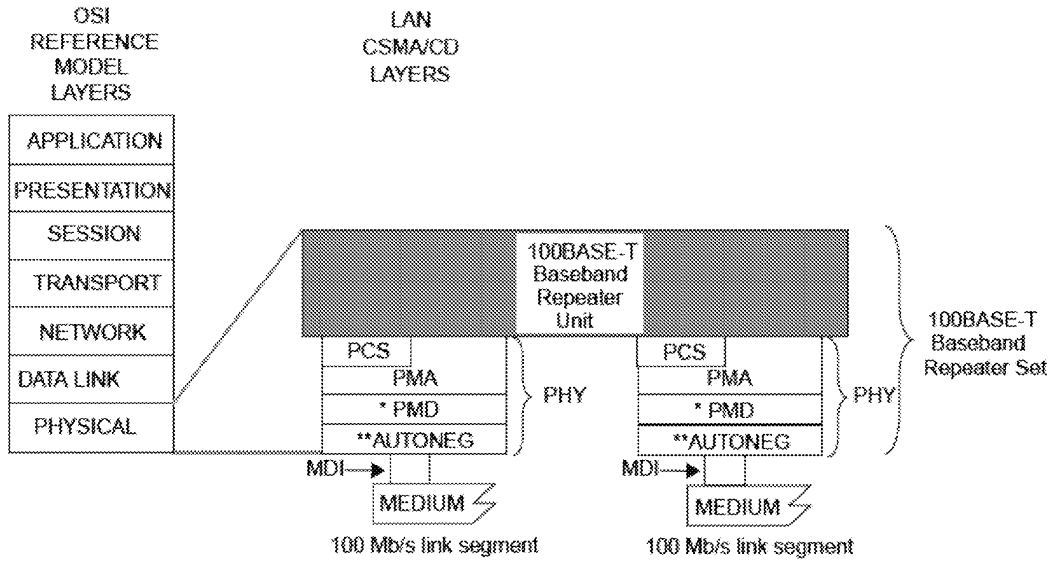
| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| PD1 | Compliance with 100BASE-X PMD Service Interface | 26.1 | M | | See 24.2.3 |
| PD2 | Compliance with ISO 9314-3: 1990 8, 9, and 10 | 26.4 | M | | |
| PD3 | Precedence over ISO 9314-3: 1990 | 26.4 | M | | |
| PD4 | MIC receptacle keying | 26.4.2 | FMC: M | | "M" |
| PD5 | Crossover function in cable | 26.4.3 | M | | |

# 27. Repeater for 100 Mb/s baseband networks

## 27.1 Overview

### 27.1.1 Scope

Clause 27 defines the functional and electrical characteristics of a repeater for use with 100BASE-T 100 Mb/s baseband networks. A repeater for any other ISO/IEC 8802-3 network type is beyond the scope of this clause. The relationship of this standard to the entire ISO/IEC 8802-3 CSMA/CD LAN standard is shown in figure 27-1. The purpose of the repeater is to provide a simple, inexpensive, and flexible means of coupling two or more segments.



MDI = MEDIUM DEPENDENT INTERFACE
MII = MEDIA INDEPENDENT INTERFACE

PCS = PHYSICAL CODING SUBLAYER
PMA = PHYSICAL MEDIUM ATTACHMENT
PHY = PHYSICAL LAYER DEVICE
PMD = PHYSICAL MEDIUM DEPENDENT

* PMD is specified for 100BASE-TX and -FX only; 100BASE-T4 does not use this layer.
Use of MII between PCS and baseband repeater unit is optional.
** AUTONEG is optional.

**Figure 27-1—100BASE-T repeater set relationship to the OSI reference model**

#### 27.1.1.1 Repeater set

Repeater sets are an integral part of all 100 Mb/s baseband networks with more than two DTEs and are used to extend the physical system topology by providing a means of coupling two or more segments. Multiple repeater sets are permitted within a single collision domain to provide the maximum connection path length. Segments may be connected directly by a repeater or a pair of repeaters that are, in turn, connected by a inter-repeater link (IRL). Allowable topologies shall contain only one operative signal path between any two points on the network. A repeater set is not a station and does not count toward the overall limit of 1024 stations on a network.

A repeater set can receive, and if necessary decode, data from any segment under worst-case noise, timing, and signal amplitude conditions. It retransmits the data to all other segments attached to it with timing, amplitude, and, if necessary, coding restored. The retransmission of data occurs simultaneously with reception. If a collision occurs, the repeater set propagates the collision event throughout the network by transmitting a Jam signal. A repeater set also provides a degree of protection to a network by isolating a faulty segment's carrier activity from propagating through the network.

#### 27.1.1.2 Repeater unit

A repeater unit is a subset of a repeater set containing all the repeater-specific components and functions, exclusive of PHY components and functions. A repeater unit connects to the PMA and, if necessary, the PCS sublayers of its PHYs.

#### 27.1.1.3 Repeater classes

Two classes of repeater sets are defined—Class I and Class II.

Class I:
A type of repeater set specified such that in a maximum length segment topology, only one such repeater set may exist between any two DTEs within a single collision domain.

Class II:
A type of repeater set specified such that in a maximum length segment topology, only two such repeater sets may exist between any two DTEs within a single collision domain.

More complex topologies are possible in systems that do not use worst-case cable. See clause 29 for requirements.

### 27.1.2 Application perspective

This subclause states the broad objectives and assumptions underlying the specification defined through clause 27.

#### 27.1.2.1 Objectives

a) Provide physical means for coupling two or more LAN segments at the Physical Layer.
b) Support interoperability of independently developed physical, electrical, and optical interfaces.
c) Provide a communication channel with a mean bit error rate, at the physical service interface equivalent to that for the attached PHY.
d) Provide for ease of installation and service.
e) Ensure that fairness of DTE access is not compromised.
f) Provide for low-cost networks, as related to both equipment and cabling.
g) Make use of building wiring appropriate for the supported PHYs and telephony wiring practices.

#### 27.1.2.2 Compatibility considerations

All implementations of the repeater set shall be compatible at the MDI. The repeater set is defined to provide compatibility among devices designed by different manufacturers. Designers are free to implement circuitry within the repeater set in an application-dependent manner provided the appropriate PHY specifications are met.

#### 27.1.2.2.1 Internal segment compatibility

Implementations of the repeater set that contain a MAC layer for network management or other purposes, irrespective of whether they are connected through an exposed repeater port or are internally ported, shall conform to the requirements of clause 30 on that port if repeater management is implemented.

### 27.1.3 Relationship to PHY

A close relationship exists between clause 27 and the PHY clauses, clause 23 for the 100BASE-T4 PHY and clauses 24 to 26 for the 100BASE-X PHYs. The PHY's PMA, PCS, and MDI specification provide the actual medium attachment, including drivers, receivers, and Medium Interface Connectors for the various supported media. The repeater clause does not define a new PHY; it utilizes the existing PHYs complete and without modification.

## 27.2 PMA interface messages

The messages between the repeater unit and the PMA in the PHY utilizes the PMA service interface defined in 23.3 and 24.3. The PMA service interface primitives are summarized below:

PMA_TYPE.indicate
PMA_UNITDATA.request
PMA_UNITDATA.indicate
PMA_CARRIER.indicate
PMA_LINK.indicate
PMA_RXERROR.indicate

## 27.3 Repeater functional specifications

A repeater set provides the means whereby data from any segment can be received under worst case noise, timing, and amplitude conditions and then retransmitted with timing and amplitude restored to all other attached segments. Retransmission of data occurs simultaneously with reception. If a collision occurs, the repeater set propagates the collision event throughout the network by transmitting a Jam signal. If an error is received by the repeater set, no attempt is made to correct it and it is propagated throughout the network by transmitting an invalid signal.

The repeater set provides the following functional capability to handle data flow between ports:

a) *Signal restoration.* Provides the ability to restore the timing and amplitude of the received signal prior to retransmission.
b) *Transmit function.* Provides the ability to output signals on the appropriate port and encoded appropriately for that port. Details of signal processing are described in the specifications for the PHYs.
c) *Receive function.* Provides the ability to receive input signals presented to the ports. Details of signal processing are described in the specifications for the PHYs.
d) *Data-Handling function.* Provides the ability to transfer code-elements between ports in the absence of a collision.
e) *Received Event-Handling requirement.* Provides the ability to derive a carrier signal from the input signals presented to the ports.
f) *Collision-Handling function.* Provides the ability to detect the simultaneous reception of frames at two or more ports and then to propagate a Jam message to all connected ports.
g) *Error-Handling function.* Provides the ability to prevent substandard links from generating streams of false carrier and interfering with other links.

h) *Partition function.* Provides the ability to prevent a malfunctioning port from generating an excessive number of consecutive collisions and indefinitely disrupting data transmission on the network.

i) *Receive Jabber function.* Provides the ability to interrupt the reception of abnormally long streams of input data.

### 27.3.1 Repeater functions

The repeater set shall provide the Signal Restoration, Transmit, Receive, Data Handling, Received Event Handling, Collision Handling, Error Handling, Partition, and Receive Jabber functions. The repeater is transparent to all network acquisition activity and to all DTEs. The repeater will not alter the basic fairness criterion for all DTEs to access the network or weigh it toward any DTE or group of DTEs regardless of network location.

The Transmit and Receive functional requirements are specified by the PHY clauses, clause 23 for 100BASE-T4 and clauses 24 to 26 for 100BASE-X.

#### 27.3.1.1 Signal restoration functional requirements

##### 27.3.1.1.1 Signal amplification

The repeater set (including its integral PHYs) shall ensure that the amplitude characteristics of the signals at the MDI outputs of the repeater set are within the tolerances of the specification for the appropriate PHY type. Therefore, any loss of signal-to-noise ratio due to cable loss and noise pickup is regained at the output of the repeater set as long as the incoming data is within system specification.

##### 27.3.1.1.2 Signal wave-shape restoration

The repeater set (including its integral PHYs) shall ensure that the wave-shape characteristics of the signals at the MDI outputs of a repeater set are within the specified tolerance for the appropriate PHY type. Therefore, any loss of wave-shape due to PHYs and media distortion is restored at the output of the repeater set.

##### 27.3.1.1.3 Signal retiming

The repeater set (including its integral PHYs) shall ensure that the timing of the encoded data output at the MDI outputs of a repeater set are within the specified tolerance for the appropriate PHY type. Therefore, any receive jitter from the media is removed at the output of the repeater set.

#### 27.3.1.2 Data-handling functional requirements

##### 27.3.1.2.1 Data frame forwarding

The repeater set shall ensure that the data frame received on a single input port is distributed to all other output ports in a manner appropriate for the PHY type of that port. The data frame is that portion of the packet after the SFD and before the end-of-frame delimiter. The only exceptions to this rule are when contention exists among any of the ports, when the receive port is partitioned as defined in 27.3.1.6, when the receive port is in the Jabber state as defined in 27.3.1.7, or when the receive port is in the Link Unstable state as defined in 27.3.1.5.1. Between unpartitioned ports, the rules for collision handling (see 27.3.1.4) take precedence.

##### 27.3.1.2.2 Received code violations

The repeater set shall ensure that any code violations received while forwarding a packet are propagated to all outgoing segments. These code violations shall be forwarded as received or replaced by bad_code (see 23.2.1.2) or /H/ (see 24.2.2.1) code-groups, as appropriate for the outgoing PHY type. Once a received code violation has been replaced by bad_code or the /H/ code-group, this substitution shall continue for the remainder of the packet regardless of its content. The only exception to this rule is when contention exists among any of the ports, where the rules for collision handling (see 27.3.1.4) then take precedence.

#### 27.3.1.3 Received event-handling functional requirements

##### 27.3.1.3.1 Received event handling

For all its ports, the repeater set shall implement a function (scarrier_present) that represents a received event. Received events include both the data frame and any encapsulation of the data frame such as Preamble, SFD and the code-groups /H/, /J/, /K/, bad_code, eop, /T/, /R/, etc. A received event is exclusive of the IDLE pattern. Upon detection of scarrier_present from one port, the repeater set repeats all received signals in the data frame from that port to the other port (or ports) as described in figure 27-2.

##### 27.3.1.3.2 Preamble regeneration

The repeater set shall output preamble as appropriate for the outgoing PHY type followed by the SFD.

##### 27.3.1.3.3 Start-of-packet propagation delay

The start-of-packet propagation delay for a repeater set is the time delay between the start of the packet (see 24.6 and 23.11.3) on its repeated-from (input) port to the start of the packet on its repeated-to (output) port (or ports). This parameter is referred to as the SOP delay. The maximum value of this delay is constrained by table 27-2.

##### 27.3.1.3.4 Start-of-packet variability

The start-of-packet variability for a repeater set is defined as the total worst-case difference between start-of-packet propagation delays for successive packets separated by 104 bit times (BT) or less at the same input port. The variability shall be less than or equal to those specified in table 27-1.

**Table 27-1—Start-of-packet variability**

| Input port type | Variability (BT) |
|---|---|
| 100BASE-FX | 7.0 |
| 100BASE-TX | 7.0 |
| 100BASE-T4 | 8.0 |

#### 27.3.1.4 Collision-handling functional requirements

##### 27.3.1.4.1 Collision detection

The repeater performs collision detection by monitoring all its enabled input ports for received events. When the repeater detects received events on more than one input port, it shall enter a collision state and transmit the Jam message to all of its output ports.

#### 27.3.1.4.2 Jam generation

While a collision is occurring between any of its ports, the repeater unit shall transmit the Jam message to all of the PMAs to which it is connected. The Jam message shall be transmitted in accordance with the repeater state diagram in figure 27-4 and figure 27-5.

#### 27.3.1.4.3 Collision-jam propagation delay

The start-of-collision Jam propagation delay for a repeater set is the time delay between the start of the second packet input signals to arrive at its port and the start of Jam (see 24.6 and 23.11) out on all ports. This parameter is referred to as the SOJ delay. The delay shall be constrained by table 27-2.

**Table 27-2—Start-of-packet propagation and start-of-collision Jam propagation delays**

| Class I repeater | Class II repeater with all ports TX/FX | Class II repeater with any port T4 |
|---|---|---|
| SOP + SOJ ″ 140 BT | SOP ″ 46 BT, SOJ ″ 46 BT | SOP+SOJ ″ 67 BT |

#### 27.3.1.4.4 Cessation-of-collision Jam propagation delay

The cessation-of-collision Jam propagation delay for a repeater set is the time delay between the end of the packet (see 24.6 and 23.11.3) that creates a state such that Jam should end at a port and the end of Jam (see 24.6 and 23.11.3) at that port. The states of the input signals that should cause Jam to end are covered in detail in the repeater state diagrams. This parameter is referred to as the EOJ delay. The delay shall be constrained by table 27-3.

**Table 27-3—Cessation-of-collision Jam propagation delay**

| Class I repeater | Class II repeater |
|---|---|
| EOJ ″ SOP | EOJ ″ SOP |

### 27.3.1.5 Error-handling functional requirements

#### 27.3.1.5.1 100BASE-X carrier integrity functional requirements

In 100BASE-TX and 100BASE-FX systems, it is desirable that the repeater set protect the network from some transient fault conditions that would disrupt network communications. Potential likely causes of such conditions are DTE and repeater power-up and power-down transients, cable disconnects, and faulty wiring.

Each 100BASE-TX and 100BASE-FX repeater PMA interface shall contain a self-interrupt capability, as described in figure 27-9, to prevent a segment's spurious carrier activity from reaching the repeater unit and hence propagating through the network.

The repeater PMA interface shall count consecutive false carrier events. A false carrier event is defined as a carrier event that does not begin with a valid start-of-stream delimiter (see 24.2.2.1.4). The count shall be incremented on each false carrier event and shall be reset on reception of a valid carrier event. In addition, each PMA interface shall contain a false carrier timer, which is enabled at the beginning of a false carrier event and reset at the conclusion of such an event. A repeater unit shall transmit the Jam message to all of the PMAs to which it is connected for the duration of the false carrier event or until the duration of the event exceeds the time specified by the false_carrier_timer (see 27.3.2.1.4), whichever is shorter. The Jam message shall be transmitted in accordance with the repeater state diagram in figure 27-4 and figure 27-5. The LINK UNSTABLE condition shall be detected when the False Carrier Count exceeds the value FCCLimit (see 27.3.2.1.1) or the duration of a false carrier event exceeds the time specified by the false_carrier_timer. In addition, the LINK UNSTABLE condition shall be detected upon power-up reset.

Upon detection of LINK UNSTABLE, the port shall perform the following:

a) Inhibit sending further messages to the repeater unit.
b) Inhibit sending further output messages from the repeater unit.
c) Continue to monitor activity on that PMA interface.

The repeater shall exit the LINK UNSTABLE condition when one of the following is met:

a) The repeater has detected no activity (Idle) for more than the time specified by ipg_timer plus idle_timer (see 27.3.2.1.4) on port X.
b) A valid carrier event with a duration greater than the time specified by valid_carrier_timer (see 27.3.2.1.4) has been received, preceded by no activity (Idle) for more than the time specified by ipg_timer (see 27.3.2.1.4) on port X.

#### 27.3.1.5.2 Speed handling

If the PHY has the capability of detecting speeds other than 100 Mb/s, then the repeater set shall have the capability of blocking the flow of non-100 Mb/s signals. The incorporation of 100 Mb/s and 10 Mb/s repeater functionality within a single repeater set is beyond the scope of this standard.

### 27.3.1.6 Partition functional requirements

In large multisegment networks it may be desirable that the repeater set protect the network from some fault conditions that would disrupt network communications. A potentially likely cause of this condition could be due to a cable fault.

Each repeater PMA interface shall contain a self-interrupt capability, as described in figure 27-8, to prevent a faulty segment's carrier activity from reaching the repeater unit and hence propagating through the network. The repeater PMA interface shall count consecutive collisions. The count shall be incremented on each transmission that suffers a collision and shall be reset on a successful transmission. If this count exceeds the value CCLimit (see 27.3.2.1.1) the Partition condition shall be detected.

Upon detection of Partition, the port shall perform the following:

a) Inhibit sending further input messages to the repeater unit.
b) Continue to output messages from the repeater unit.
c) Continue to monitor activity on that PMA interface.

The repeater shall reset the Partition function when one of the following conditions is met:

a) On power-up reset.
b) The repeater has detected activity on the port for more than the number of bits specified for no_collision_timer (see 27.3.2.1.4) without incurring a collision.

### 27.3.1.7 Receive jabber functional requirements

Each repeater PMA interface shall contain a self-interrupt capability, as described in figure 27-7, to prevent an illegally long reception of data from reaching the repeater unit. The repeater PMA interface shall provide a window of duration jabber_timer bit times (see 27.3.2.1.4) during which the input messages may be passed on to other repeater unit functions. If a reception exceeds this duration, the jabber condition shall be detected.

Upon detection of jabber, the port shall perform the following:

a) Inhibit sending further input messages to the repeater unit.
b) Inhibit sending further output messages from the repeater unit.

The repeater PMA interface shall reset the Jabber function and re-enable data transmission and reception when either one of the following conditions is met:

a) On power-up reset.
b) When carrier is no longer detected.

### 27.3.2 Detailed repeater functions and state diagrams

A precise algorithmic definition is given in this subclause, providing a complete procedural model for the operation of a repeater, in the form of state diagrams. Note that whenever there is any apparent ambiguity concerning the definition of repeater operation, the state diagrams should be consulted for the definitive statement.

The model presented in this subclause is intended as a primary specification of the functions to be provided by any repeater unit. It is important to distinguish, however, between the model and a real implementation. The model is optimized for simplicity and clarity of presentation, while any realistic implementation should place heavier emphasis on such constraints as efficiency and suitability to a particular implementation technology.

It is the functional behavior of any repeater unit implementation that shall match the standard, not the internal structure. The internal details of the procedural model are useful only to the extent that they help specify the external behavior clearly and precisely. For example, the model uses a separate Receive Port Jabber state diagram for each port. However, in actual implementation, the hardware may be shared.

The notation used in the state diagram follows the conventions of 1.2.1. Note that transitions shown without source states are evaluated at the completion of every state and take precedence over other transition conditions.

### 27.3.2.1 State diagram variables

#### 27.3.2.1.1 Constants

CCLimit

The number of consecutive collisions that must occur before a segment is partitioned.

Values: Positive integer greater than 60.

FCCLimit

The number of consecutive False Carrier events that must occur before a segment is isolated.

Value: 2.

#### 27.3.2.1.2 Variables

activity(Port designation)

Indicates port activity status. The repeater core effects a summation of this variable received from all its attached ports and responds accordingly.

Values:
0; no frame or packet activity at any port.
1; exactly 1 port of the repeater set has frame or packet activity input.
>1; more than 1 port of the repeater set has frame or packet activity input. Alternately, one or more ports has detected a carrier that is not valid.

all_data_sent

Indicates if all received data frame bits or code-groups from the current frame have been sent. During or after collision the all_data_sent variable follows the inverse of the carrier of port N.

Values:
true; all received data frame bits or code-groups have been sent.
false; all received data frame bits or code-groups have not been sent.

begin

The Interprocess flag controlling state diagram initialization values.

Values:
true
false

carrier_status(X)

Signal received from PMA; indicates the status of sourced Carrier input at port X.

Values:
ON; the carrier_status parameter of the PMA_CARRIER.indicate primitive for port X is ON.
OFF; the carrier_status parameter of the PMA_CARRIER.indicate primitive for port X is OFF.

data_ready

Indicates if the repeater has detected and/or decoded the MAC SFD and is ready to send the received data.

Values:
true; the MAC SFD has been detected and/or decoded.
false; the MAC SFD has not been detected nor decoded.

force_jam(X)

Flag from Carrier Integrity state diagram for port X, which determines whether all ports should transmit Jam.

Values:
true; the Carrier Integrity Monitor has determined that it requires all ports be forced to transmit Jam.
false; the Carrier Integrity Monitor has determined that it does not require all ports be forced to transmit Jam.

Default: for T4 ports: false

isolate(X)

Flag from Carrier Integrity state diagram for port X, which determines whether a port should be enabled or disabled.

Values:
true; the Carrier Integrity Monitor has determined the port should be disabled.
false; the Carrier Integrity Monitor has determined the port should be enabled.

jabber(X)

Flag from Receive Timer state diagram for port X which indicates that the port has received excessive length activity.

Values:
true; port has exceeded the continuous activity limit.
false; port has not exceeded the continuous activity limit.

link_status(X)

Signal received from PMA; indicates link status for port X (see 23.1.4.5 and 24.3.1.5).

Values: OK; the link_status parameter of the PMA_LINK.indicate primitive for port X is OK.
READY; the link_status parameter of the PMA_LINK.indicate primitive for port X is READY.
FAIL; the link_status parameter of the PMA_LINK.indicate primitive for port X is FAIL.

opt(X)

Implementation option. Either value may be chosen for repeater implementation.

Values: true; port will emit the JamT4 pattern in response to collision conditions.
false; port will append Jam pattern after preamble and SFD in response to collision conditions.

OUT(X)

Type of output repeater is sourcing at port X.

Values: Idle; repeater is transmitting an IDLE pattern as described by 23.4.1.2 or 24.2.2.1.2.
In(N); repeater is transmitting rx_code_bit(s) as received from port (N) except /J/K/ (see 24.3.4.2).
Pream; repeater is sourcing preamble pattern as defined by the PMA or PCS of the port type (see 23.2.1.2, 24.2.2.2, figure 23-6, and figure 24-5).
Data; repeater is transmitting data frame on port X. This data represents the original MAC source data field, properly encoded for the PHY type (see 23.2.1.2 and 24.2.2.2).
Jam; repeater is sourcing well formed arbitrary data encodings, excluding SFD, to the port PMA.
JamX; repeater is sourcing the pattern 010101... repetitively on port X.
JamT4; repeater is sourcing the pattern +–+–... repetitively on port X
SFD; repeater is sourcing the Start Frame Delimiter on port X encoded as defined by the appropriate PHY (see 23.2.3 and figure 24-5).
/J/K/; repeater is sourcing the code-groups /J/K/ as defined by the PMA on port X (see 24.2.2.1.4).
/T/R/; repeater is sourcing the code-groups /T/R/ as defined by the PMA on port X (see 24.2.2.1.5).
DF; repeater is sourcing the data frame of the packet on port X. These are code elements originating on port N exclusive of EOP1-5, SOSA, and SOSB (see 23.2.3 and 23.2.4).
EOP; repeater is sourcing end-of-packet delimiter (EOP1-5) as defined by the appropriate PMA on port X (see 23.2.1.2 and 23.2.4.1).
bad_code; repeater is sourcing bad_code as defined by the PMA of the transmit port (see 23.2.4.1).
tx_err; repeater is sourcing a transmit error code element, either bad_code (see 23.2.4.1) or the code-group /H/ (see 24.2.2.1) as appropriate to the outgoing PHY type.

partition(X)

Flag from Partition state diagram for port X, which determines whether a port receive path should be enabled or disabled.

Values: true; port has exceeded the consecutive collision limit.
false; port has not exceeded the consecutive collision limit.

rxerror_status(X)

Signal received from PMA; indicates if port X has detected an error condition from the PMA (see 23.3.7.1 and figure 24-14). The repeater need not propagate this error condition during collision events.

Values: ERROR; the rxerror_status parameter of the PMA_RXERROR.indicate primitive for port X is ERROR.
NO_ERROR; the rxerror_status parameter of the PMA_RXERROR.indicate primitive for port X is NO_ERROR.

RX_ER(X)

Signal received from PCS; indicates if port X has detected an error condition from the PCS (see 23.2.1.4, 24.2.3.2, figure 23-10, and figure 24-11). The repeater need not propagate this error condition during collision events.

Values: true; the PCS RX_ER signal for port X is asserted.
false; the PCS RX_ER signal for port X is negated.

scarrier_present(X)

Signal received from PMA; indicates the status of sourced Carrier input at port X.

Values: true; the carrier_status parameter of the PMA_CARRIER.indicate primitive for port X is ON.
false; the carrier_status parameter of the PMA_CARRIER.indicate primitive for port X is OFF.

source_type(X)

Signal received from PMA; indicates PMA type for port X. The first port to assert activity maintains the source type status for all transmitting port(s) until activity is deasserted. Repeaters may optionally force nonequality on comparisons using this variable. It must then follow the behavior of the state diagrams accordingly and meet all the delay parameters as applicable for the real implemented port type(s).

Values: FXTX; the pma_type parameter of the PMA_TYPE.indicate primitive for port X is X.
T4; the pma_type parameter of the PMA_TYPE.indicate primitive for port X is T4.

#### 27.3.2.1.3 Functions

command(X)

A function that passes an inter-process flag to all ports specified by X.

Values: copy; indicates that the repeater core has summed the activity levels of its active ports and is in the ACTIVE state.
collision; indicates that the repeater core has summed the activity levels of its active ports and is in the JAM state.
quiet; indicates that the repeater core has summed the activity levels of its active ports and is in the IDLE state.

port(Test)

A function that returns the designation of a port passing the test condition. For example, port(activity = scarrier_present) returns the designation: X for a port for which scarrier_present = true. If multiple ports meet the test condition, the Port function will be assigned one and only one of the acceptable values.

#### 27.3.2.1.4 Timers

All timers operate in the same fashion. A timer is reset and starts timing upon entering a state where "start x_timer" is asserted. At time "x" after the timer has been started, "x_timer_done" is asserted and remains asserted until the timer is reset. At all other times, "x_timer_not_done" is asserted.

When entering a state where "start x_timer" is asserted, the timer is reset and restarted even if the entered state is the same as the exited state.

The timers used in the repeater state diagrams are defined as follows:

false_carrier_timer

Timer for length of false carrier (27.3.1.5.1) that must be present before the ISOLATION state is entered. The timer is done when it reaches 450 – 500 BT.

idle_timer

Timer for length of time without carrier activity that must be present before the ISOLATION state is exited (27.3.1.5.1). The timer is done when it reaches 33 000 ± 25% BT.

ipg_timer

Timer for length of time without carrier activity that must be present before carrier integrity tests (27.3.1.5.1) are re-enabled. The timer is done when it reaches 64 – 86 BT.

jabber_timer

Timer for length of carrier which must be present before the Jabber state is entered (27.3.1.7). The timer is done when it reaches 40 000 – 75 000 BT.

no_collision_timer

Timer for length of packet without collision before the Partition state is exited (27.3.1.6). The timer is done when it reaches 450 – 560 BT.

valid_carrier_timer

Timer for length of valid carrier that must be present before the Isolation state is exited (27.3.1.5.1). The timer is done when it reaches 450 – 500 BT.

#### 27.3.2.1.5 Counters

CC(X)

Consecutive port collision count for port X. Partitioning occurs on a terminal count of CCLimit being reached.

Values: Non-negative integers up to a terminal count of CCLimit.

FCC(X)

False Carrier Counter for port X. Isolation occurs on a terminal count of FCCLimit being reached.

Values: Non-negative integers up to a terminal count of FCCLimit.

#### 27.3.2.1.6 Port designation

Ports are referred to by number. Port information is obtained by replacing the X in the desired function with the number of the port of interest. Ports are referred to in general as follows:

X

Generic port designator. When X is used in a state diagram, its value is local to that diagram and not global to the set of state diagrams.

N

Is defined by the Port function on exiting the IDLE or JAM states of figure 27-2. It indicates a port that caused the exit from these states.

ALL

Indicates all repeater ports are to be considered. All ports shall meet test conditions in order for the test to pass.

ALLXN

Indicates all ports except N should be considered. All ports considered shall meet the test conditions in order for the test to pass.

ANY

Indicates all ports are to be considered. One or more ports shall meet the test conditions in order for the test to pass.

### 27.3.2.2 State diagrams

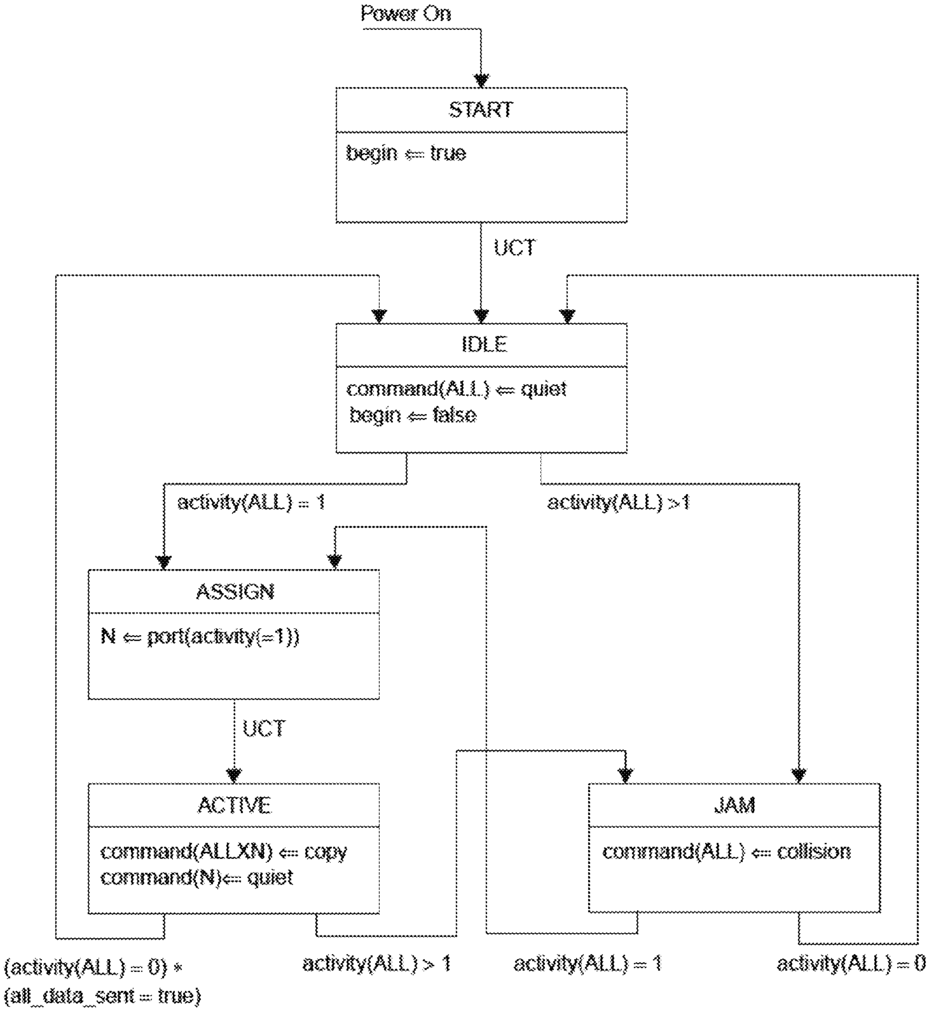


Figure 27-2—Repeater core state diagram

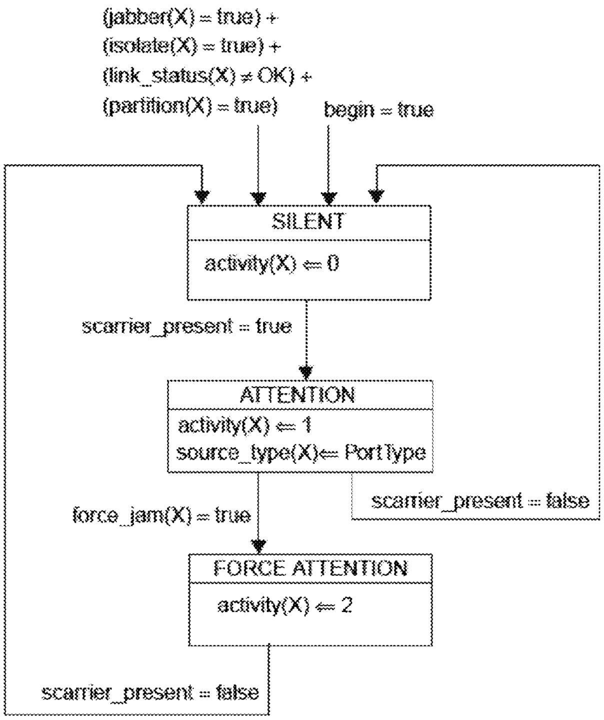


Figure 27-3—Receive state diagram for port X

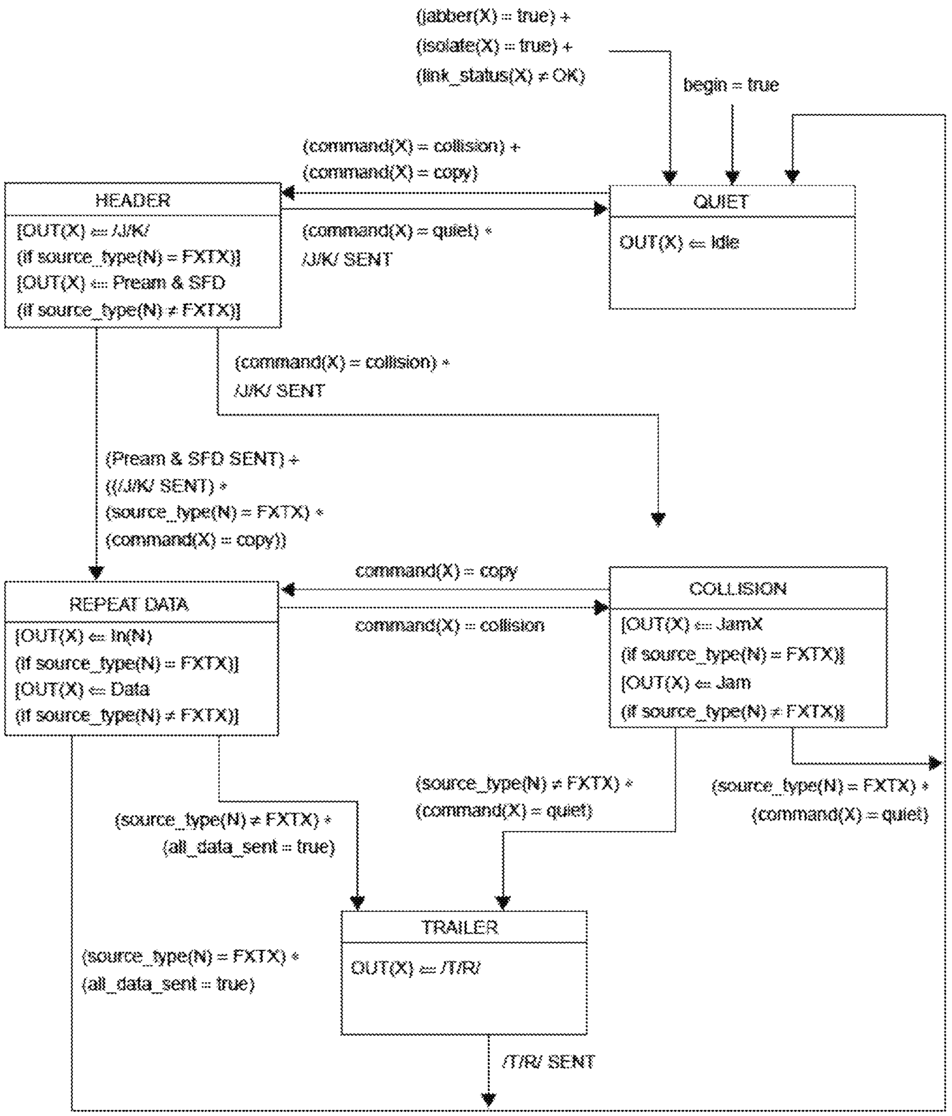


**Figure 27-4—100BASE-TX and 100BASE-FX transmit state diagram for port X**

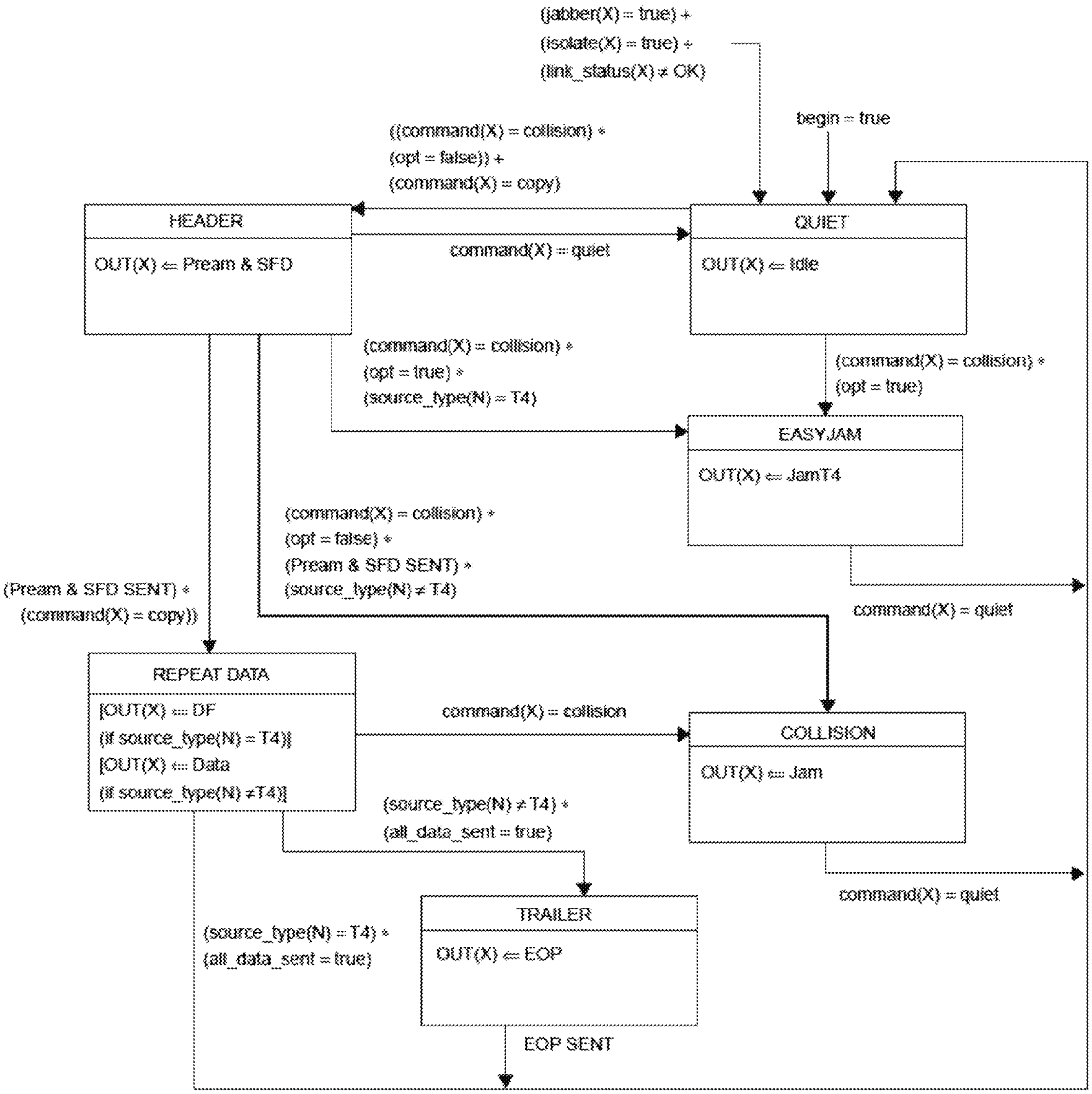


**Figure 27-5—100BASE-T4 transmit state diagram for port X**

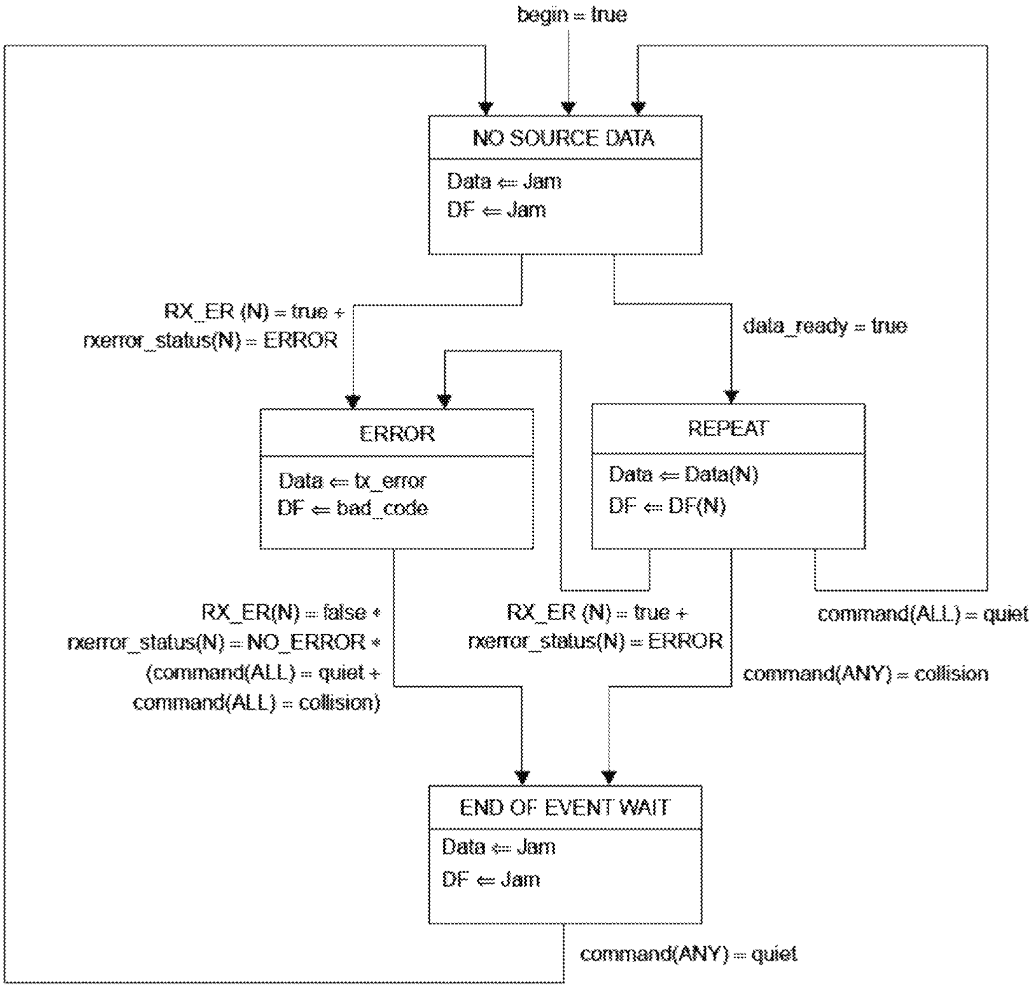


Figure 27-6—Repeater data-handler state diagram

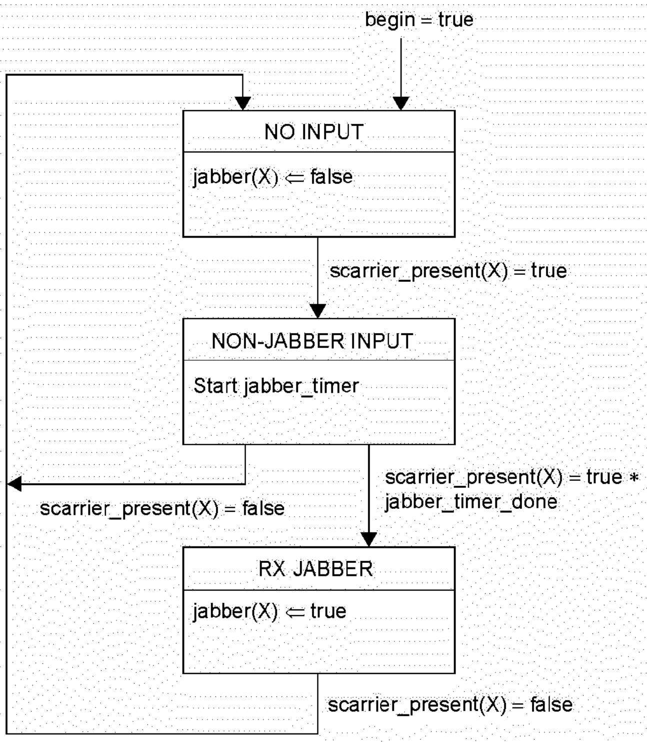


Figure 27-7—Receive timer state diagram for port X

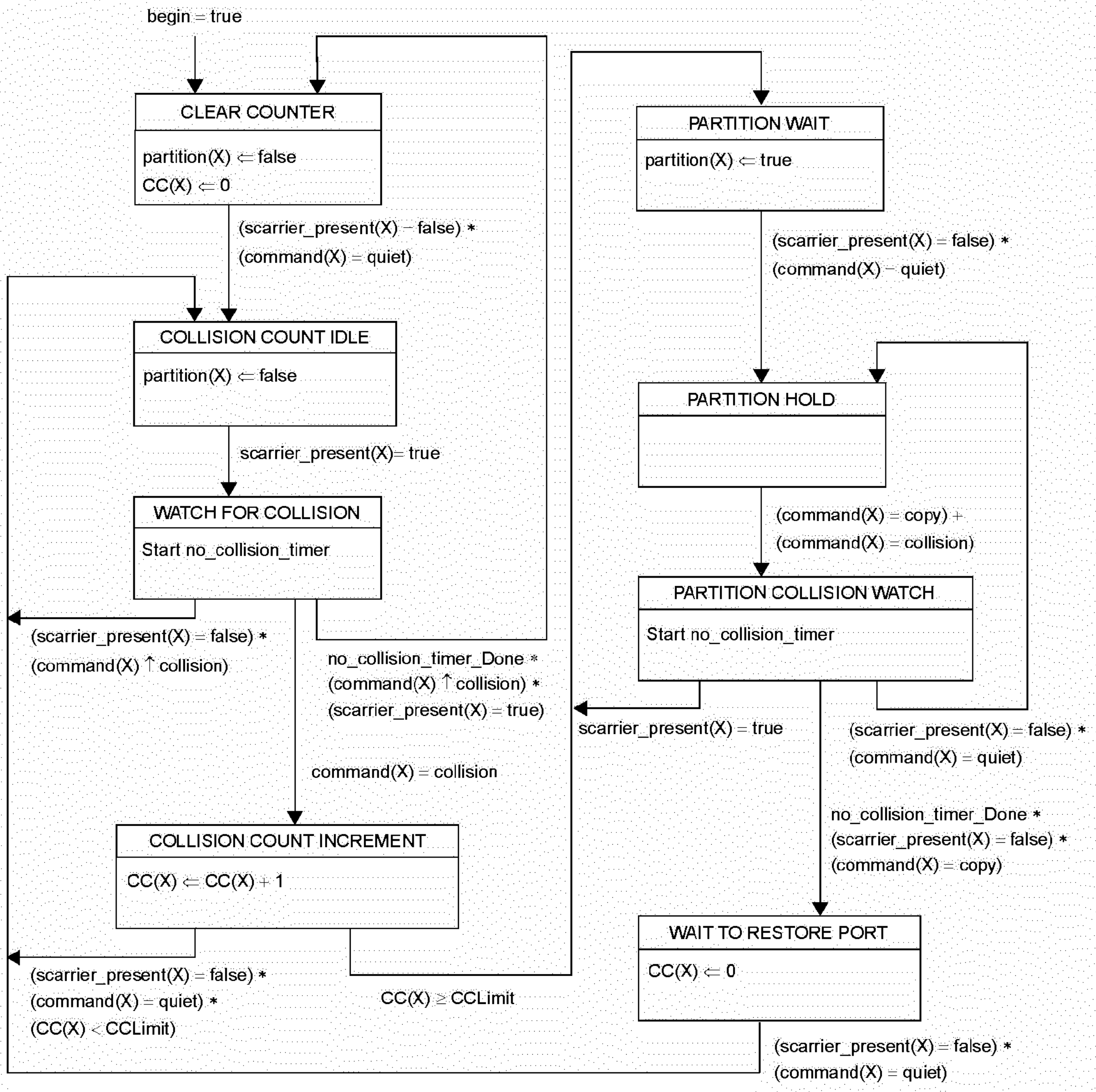


Figure 27-8—Partition state diagram for port X

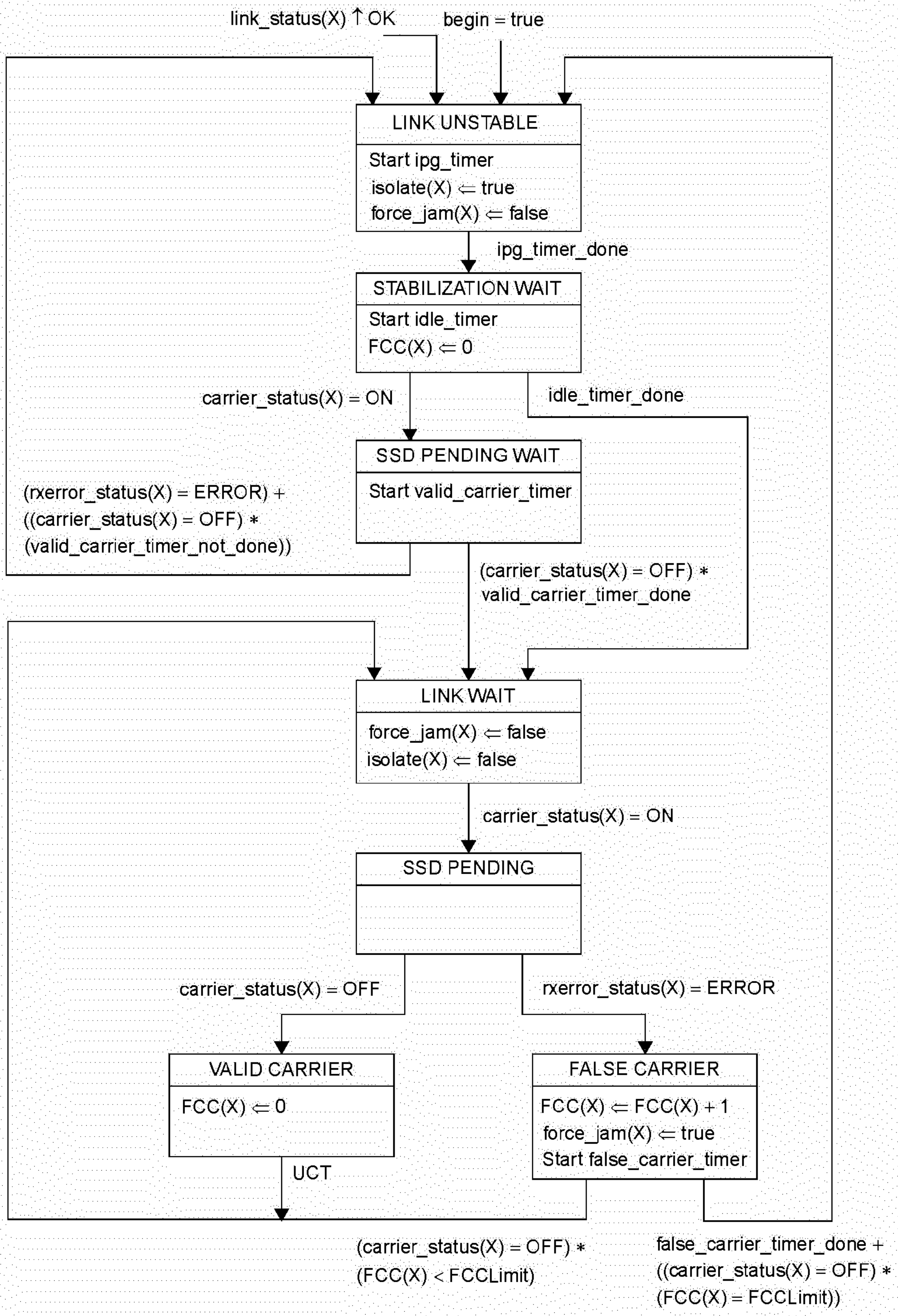


**Figure 27-9—100BASE-X carrier integrity monitor state diagram for port X**

## 27.4 Repeater electrical specifications

### 27.4.1 Electrical isolation

Network segments that have different isolation and grounding requirements shall have those requirements provided by the port-to-port isolation of the repeater set.

## 27.5 Environmental specifications

### 27.5.1 General safety

All equipment meeting this standard shall conform to IEC 950: 1991.

### 27.5.2 Network safety

This subclause sets forth a number of recommendations and guidelines related to safety concerns; the list is neither complete nor does it address all possible safety issues. The designer is urged to consult the relevant local, national, and international safety regulations to ensure compliance with the appropriate requirements.

LAN cable systems described in this subclause are subject to at least four direct electrical safety hazards during their installation and use. These hazards are as follows:

a) Direct contact between LAN components and power, lighting, or communications circuits.
b) Static charge buildup on LAN cables and components.
c) High-energy transients coupled onto the LAN cable system.
d) Voltage potential differences between safety grounds to which the various LAN components are connected.

Such electrical safety hazards must be avoided or appropriately protected against for proper network installation and performance. In addition to provisions for proper handling of these conditions in an operational system, special measures must be taken to ensure that the intended safety features are not negated during installation of a new network or during modification or maintenance of an existing network. Isolation requirements are defined in 27.5.3.

#### 27.5.2.1 Installation

Sound installation practice, as defined by applicable local codes and regulations, shall be followed in every instance in which such practice is applicable.

#### 27.5.2.2 Grounding

The safety ground, or chassis ground for the repeater set, shall be provided through the main ac power cord via the third wire ground as defined by applicable local codes and regulations. It is recommended that an external PHY to the repeater should also be mechanically grounded to the repeater unit through the power and ground signals in the MII connection and via the metal shell and shield of the MII connector if available.

If the MDI connector should provide a shield connection, the shield may be connected to the repeater safety ground. A network segment connected to the repeater set through the MDI may use a shield. If both ends of the network segment have a shielded MDI connector available, then the shield may be grounded at both ends according to local regulations and ISO/IEC 11801: 1995, and as long as the ground potential difference between both ends of the network segment is less than 1 V rms. The same rules apply towards an inter-repeater link between two repeaters. Multiple repeaters should reside on the same power main; if not, then it is highly recommended that the repeaters be connected via fiber.

WARNING—It is assumed that the equipment to which the repeater is attached is properly grounded and not left floating nor serviced by a "doubly insulated ac power distribution system." The use of floating or insulated equipment, and the consequent implications for safety, are beyond the scope of this standard.

#### 27.5.2.3 Installation and maintenance guidelines

During installation and maintenance of the cable plant, care should be taken to ensure that uninsulated network cable connectors do not make electrical contact with unintended conductors or ground.

### 27.5.3 Electrical isolation

There are two electrical power distribution environments to be considered that require different electrical isolation properties:

a) *Environment A.* When a LAN or LAN segment, with all its associated interconnected equipment, is entirely contained within a single low-voltage power distribution system and within a single building.
b) *Environment B.* When a LAN crosses the boundary between separate power distribution systems or the boundary of a single building.

#### 27.5.3.1 Environment A requirements

Attachment of network segments via repeater sets requires electrical isolation of 500 V rms, one-minute withstand, between the segment and the protective ground of the repeater unit.

#### 27.5.3.2 Environment B requirements

The attachment of network segments that cross environment B boundaries requires electrical isolation of 1500 V rms, one-minute withstand, between each segment and all other attached segments and also the protective ground of the repeater unit.

The requirements for interconnected electrically conducting LAN segments that are partially or fully external to a single building environment may require additional protection against lightning strike hazards. Such requirements are beyond the scope of this standard. It is recommended that the above situation be handled by the use of nonelectrically conducting segments (e.g., fiber optic).

It is assumed that any nonelectrically conducting segments will provide sufficient isolation within that media to satisfy the isolation requirements of environment B.

### 27.5.4 Reliability

A two-port repeater set shall be designed to provide a mean time between failure (MTBF) of at least 50 000 hours of continuous operation without causing a communications failure among stations attached to the network medium. Repeater sets with more than two ports shall add no more than $3.46 \times 10^{-6}$ failures per hour for each additional port.

The repeater set electronics should be designed to minimize the probability of component failures within the repeater electronics that prevent communications among other PHYs on the individual segments. Connectors and other passive components comprising the means of connecting the repeater to the cable should be designed to minimize the probability of total network failure.

### 27.5.5 Environment

#### 27.5.5.1 Electromagnetic emission

The repeater shall comply with applicable local and national codes for the limitation of electromagnetic interference.

#### 27.5.5.2 Temperature and humidity

The repeater is expected to operate over a reasonable range of environmental conditions related to temperature, humidity, and physical handling (such as shock and vibration). Specific requirements and values for these parameters are considered to be beyond the scope of this standard.

It is recommended that manufacturers indicate in the literature associated with the repeater the operating environmental conditions to facilitate selection, installation, and maintenance.

## 27.6 Repeater labeling

It is required that each repeater (and supporting documentation) shall be labeled in a manner visible to the user with these parameters:

a) Crossover ports appropriate to the respective PHY should be marked with an X.
b) The repeater set class type should be labeled in the following manner:
    1) Class I: a Roman numeral "I" centered within a circle.
    2) Class II: a Roman numeral "II" centered within a circle.

Additionally it is recommended that each repeater (and supporting documentation) also be labeled in a manner visible to the user with at least these parameters:

a) Data rate capability in Mb/s
b) Any applicable safety warnings
c) Port type, i.e., 100BASE-TX and 100BASE-T4
d) Worst-case bit time delays between any two ports appropriate for
    1) Start-of-packet propagation delay
    2) Start-of-collision Jam propagation delay
    3) Cessation-of-collision Jam propagation delay

## 27.7 Protocol Implementation Conformance Statement (PICS) proforma for clause 27, Repeater for 100 Mb/s baseband networks[26]

### 27.7.1 Introduction

The supplier of a protocol implementation that is claimed to conform to IEEE Std 802.3u-1995, Repeater for 100 Mb/s baseband networks, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

### 27.7.2 Identification

### 27.7.2.1 Implementation identification

| | |
|---|---|
| Supplier | |
| Contact point for enquiries about the PICS | |
| Implementation Name(s) and Version(s) | |
| Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s) | |
| NOTES<br>1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.<br>2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model). | |

### 27.7.2.2 Protocol summary

| | |
|---|---|
| Identification of protocol standard | IEEE Std 802.3u-1995, Repeater for 100 Mb/s baseband networks |
| Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS | |
| Have any Exception items been required? No [ ] Yes [ ]<br>(See clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3u-1995.) | |
| Date of Statement | |

---

[26] *Copyright release for PICS proformas* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

### 27.7.3 Major capabilities/options

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| *FXP | Repeater supports 100BASEFX connections | 27.1.2.2 | O | | |
| *TXP | Repeater supports 100BASETX connections | 27.1.2.2 | O | | |
| *T4P | Repeater supports 100BASET4 connections | 27.1.2.2 | O | | |
| *CLI | Repeater meets Class I delays | 27.1.1.3 | O | | |
| *CLII | Repeater meets Class II delays | 27.1.1.3 | O | | |
| *PHYS | PHYs capable of detecting non 100BASE-T signals | 27.3.1.5.2 | O | | |

In addition, the following predicate name is defined for use when different implementations from the set above have common parameters:

*XP:FXP or TXP

### 27.7.4 PICS proforma tables for the Repeater for 100 Mb/s baseband networks

#### 27.7.4.1 Compatibility considerations

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| CC1 | 100BASE-FX port compatible at the MDI | 27.1.2.2 | FXP:M | | |
| CC2 | 100BASE-TX port compatible at the MDI | 27.1.2.2 | TXP:M | | |
| CC3 | 100BASE-T4 port compatible at the MDI | 27.1.2.2 | T4P:M | | |
| CC4 | Internal segment compatibility | 27.1.2.2.1 | M | | Internal port meets clause 29 when repeater management implemented |

### 27.7.4.2 Repeater functions

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| RF1 | Signal Restoration | 27.3.1 | M | | |
| RF2 | Data Handling | 27.3.1 | M | | |
| RF3 | Received Event Handling | 27.3.1 | M | | |
| RF4 | Collision Handling | 27.3.1 | M | | |
| RF5 | Error Handling | 27.3.1 | M | | |
| RF6 | Partition | 27.3.1 | M | | |
| RF7 | Received Jabber | 27.3.1 | M | | |

### 27.7.4.3 Signal restoration function

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| SR1 | Output amplitude as required by 100BASE-FX | 27.3.1.1.1 | FXP:M | | |
| SR2 | Output amplitude as required by 100BASE-TX | 27.3.1.1.1 | TXP:M | | |
| SR3 | Output amplitude as required by 100BASE-T4 | 27.3.1.1.1 | T4P:M | | |
| SR4 | Output signal wave-shape as required by 100BASE-FX | 27.3.1.1.2 | FXP:M | | |
| SR5 | Output signal wave-shape as required by 100BASE-TX | 27.3.1.1.2 | TXP:M | | |
| SR6 | Output signal wave-shape as required by 100BASE-T4 | 27.3.1.1.2 | T4P:M | | |
| SR7 | Output data timing as required by 100BASE-FX | 27.3.1.1.3 | FXP:M | | |
| SR8 | Output data timing as required by 100BASE-TX | 27.3.1.1.3 | TXP:M | | |
| SR9 | Output data timing as required by 100BASE-T4 | 27.3.1.1.3 | T4P:M | | |

### 27.7.4.4 Data-Handling function

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| DH1 | Data frames forwarded to all ports except receiving port | 27.3.1.2.1 | M | | |
| DH2 | Data frames transmitted as appropriate for 100BASE-FX | 27.3.1.2.1 | FXP:M | | |
| DH3 | Data frames transmitted as appropriate for 100BASE-TX | 27.3.1.2.1 | TXP:M | | |
| DH4 | Data frames transmitted as appropriate for 100BASE-T4 | 27.3.1.2.1 | T4P:M | | |
| DH5 | Code Violations forwarded to all transmitting ports | 27.3.1.2.2 | M | | |
| DH6 | Code Violations forwarded as received | 27.3.1.2.2 | O.1 | | |
| DH7 | Received Code Violation forwarded as /H/ or as received | 27.3.1.2.2 | XP:O.1 | | |
| DH8 | Received Code Violation forwarded as bad_code or as received | 27.3.1.2.2 | T4P:O.1 | | |
| DH9 | Code element substitution for remainder of packet after received Code Violation | 27.3.1.2.2 | M | | |

### 27.7.4.5 Receive Event-Handling function

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| RE1 | scarrier_present detect implemented | 27.3.1.3.1 | M | | |
| RE2 | Repeat all received signals | 27.3.1.3.1 | M | | |
| RE3 | Preamble encoded as required by 100BASE-FX | 27.3.1.3.2 | FXP:M | | |
| RE4 | Preamble encoded as required by 100BASE-TX | 27.3.1.3.2 | TXP:M | | |
| RE5 | Preamble encoded as required by 100BASE-T4 | 27.3.1.3.2 | T4P:M | | |
| RE6 | Start-of-packet propagation delay, Class I repeater | 27.3.1.3.3 | CLI:M | | |
| RE7 | Start-of-packet propagation delay, Class II repeater | 27.3.1.3.3 | CLII:M | | |

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| RE8 | Start-of-packet variability for 100BASE-FX input port | 27.3.1.3.4 | FXP:M | | 7.0 BT |
| RE8 | Start-of-packet variability for 100BASE-TX input port | 27.3.1.3.4 | TXP:M | | 7.0 BT |
| RE9 | Start-of-packet variability for 100BASE-T4 input port | 27.3.1.3.4 | T4P:M | | 8.0 BT |

### 27.7.4.6 Collision-Handling function

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| CO1 | Collision Detection | 27.3.1.4.1 | M | | Receive event on more than one port |
| CO2 | Jam Generation | 27.3.1.4.2 | M | | Transmit Jam message while collision is detected |
| CO3 | Collision-Jam Propagation delay, Class I repeater. | 27.3.1.4.3 | CLI:M | | SOP + SOJ ″ 140 BT |
| CO4 | Collision-Jam Propagation delay, Class II repeater with any port T4 | 27.3.1.4.3 | CLII:M | | SOP + SOJ ″ 67 BT |
| CO5 | Collision-Jam Propagation delay, Class II repeater, all TX/FX ports | 27.3.1.4.3 | CLII:M | | SOP ″ 46, SOJ ″ 46 BT |
| CO6 | Cessation of Collision Propagation delay, Class I repeater | 27.3.1.4.4 | CLI:M | | EOJ ″ SOP |
| CO7 | Cessation of Collision Propagation delay, Class II repeater | 27.3.1.4.4 | CLII:M | | EOJ ″ SOP |

### 27.7.4.7 Error-Handling function

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| EH1 | Carrier Integrity function implementation | 27.3.1.5.1 | XP:M | | Self-interrupt of data reception |
| EH2 | False carrier count for Link Unstable detection | 27.3.1.5.1 | XP:M | | False carrier count in excess of FCCLimit |
| EH3 | False carrier count reset | 27.3.1.5.1 | XP:M | | Count reset on valid carrier |
| EH4 | False carrier timer for Link Unstable detection | 27.3.1.5.1 | XP:M | | False carrier of length in excess of false_carrier_timer |
| EH5 | Jam message duration | 27.3.1.5.1 | XP:M | | Equals duration of false carrier event, but not greater than duration of false_carrier_timer |
| EH6 | Link Unstable detection | 27.3.1.5.1 | XP:M | | False Carrier count exceed FCCLimit or False carrier exceeds the false_carrier_timer or power-up reset |
| EH7 | Messages sent to repeater unit in Link Unstable state | 27.3.1.5.1 | XP:M | | Inhibited sending messages to repeater unit |
| EH8 | Messages sent from repeater unit in Link Unstable state | 27.3.1.5.1 | XP:M | | Inhibited sending output messages |

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| EH9 | Monitoring activity on PMA interface in Link Unstable state | 27.3.1.5.1 | XP:M |  | Continue monitoring activity at PMA interface |
| EH10 | Reset of Link Unstable state | 27.3.1.5.1 | XP:M |  | No activity for more than ipg_timer plus idle_timer or Valid carrier event of duration greater than valid_carrier_timer preceded by Idle of duration greater than ipg_timer |
| EH11 | Block flow of non-100 Mb/s signals | 27.3.1.5.2 | PHYS:M |  |  |

### 27.7.4.8 Partition function

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| PA1 | Partition function implementation | 27.3.1.6 | M | | Self-interrupt of data reception |
| PA2 | Consecutive collision count for entry into partition state | 27.3.1.6 | M | | Consecutive collision in excess of CCLimit |
| PA3 | Consecutive collision counter incrementing | 27.3.1.6 | M | | Count incremented on each transmission that suffers a collision |
| PA4 | Consecutive collision counter reset | 27.3.1.6 | M | | Count reset on successful collision |
| PA5 | Messages sent to repeater unit in Partition state | 27.3.1.6 | M | | Inhibited sending messages to repeater unit |
| PA6 | Messages sent from repeater unit in Partition state | 27.3.1.6 | M | | Continue sending output messages |
| PA7 | Monitoring activity on PMA interface in Partition state | 27.3.1.6 | M | | Continue monitoring activity at PMA interface |
| PA8 | Reset of Partition state | 27.3.1.6 | M | | Power-up reset or Detecting activity for greater than duration no_collision_timer without a collision |

### 27.7.4.9 Receive Jabber function

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| RJ1 | Receive Jabber function implementation | 27.3.1.7 | M | | Self-interrupt of data reception |
| RJ2 | Excessive receive duration timer for Receive Jabber detection | 27.3.1.7 | M | | Reception duration in excess of jabber_timer |
| RJ3 | Messages sent to repeater unit in Receive Jabber state | 27.3.1.7 | M | | Inhibit sending input messages to repeater unit |
| RJ4 | Messages sent from repeater unit in Receive Jabber state | 27.3.1.7 | M | | Inhibit sending output messages |
| RJ5 | Reset of Receive Jabber state | 27.3.1.7 | M | | Power-up reset or Carrier no longer detected |

### 27.7.4.10 Repeater state diagrams

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| SD1 | Repeater core state diagram | 27.3.2.2 | M | | Meets the requirements of figure 27-2 |
| SD2 | Receive state diagram for port X | 27.3.2.2 | M | | Meets the requirements of figure 27-3 |
| SD3 | 100BASE-TX and 100BASE-FX Transmit state diagram for port X | 27.3.2.2 | XP:M | | Meets the requirements of figure 27-4 |
| SD4 | 100BASE-T4 Transmit state diagram for port X | 27.3.2.2 | T4P:M | | Meets the requirements of figure 27-5 |
| SD5 | Repeater data-handler state diagram | 27.3.2.2 | M | | Meets the requirements of figure 27-6 |
| SD6 | Receive timer for port X state diagram | 27.3.2.2 | M | | Meets the requirements of figure 27-7 |
| SD7 | Repeater partition state diagram for port X | 27.3.2.2 | M | | Meets the requirements of figure 27-8 |
| SD8 | Carrier integrity monitor for port X state diagram | 27.3.2.2 | M | | Meets the requirements of figure 27-9 |

### 27.7.4.11 Repeater electrical

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| EL1 | Port-to-port isolation | 27.4.1 | M | | Satisfies isolation and grounding requirements for attached network segments |
| EL2 | Safety | 27.5.1 | M | | IEC 950: 1991 |
| EL3 | Installation practices | 27.5.2.1 | M | | Sound, as defined by local code and regulations |
| EL4 | Grounding | 27.5.2.2 | M | | Chassis ground provided through ac mains cord |
| EL5 | 2-port repeater set MTBF | 27.5.4 | M | | At least 50 000 hours |
| EL6 | Additional port effect on MTBF | 27.5.4 | M | | No more than $3.46 \times 10^{-6}$ increase in failures per hour |
| EL7 | Electromagnetic interference | 27.5.5.1 | M | | Comply with local or national codes |

### 27.7.4.12 Repeater labeling

| Item | Feature | Subclause | Status | Support | Value/Comment |
|---|---|---|---|---|---|
| LB1 | Crossover ports | 27.6 | M | | Marked with an X |
| LB2 | Class I repeater | 27.6 | CLI:M | | Marked with a Roman numeral I centered within a circle |
| LB3 | Class II repeater | 27.6 | CLII:M | | Marked with Roman numerals II centered within a circle |
| LB4 | Data Rate | 27.6 | O | | 100 Mb/s |
| LB5 | Safety warnings | 27.6 | O | | Any applicable |
| LB6 | Port Types | 27.6 | O | | 100BASE-FX,<br>100BASE-TX or<br>100BASE-T4 |
| LB7 | Worse-case start-of-packet propagation delay | 27.6 | O | | Value in Bit Times |
| LB8 | Worse-case start-of-collision-Jam propagation delay | 27.6 | O | | Value in Bit Times |
| LB9 | Worse-case Cessation-of-Collision Jam propagation delay | 27.6 | O | | Value in Bit Times |

# 28. Physical Layer link signaling for 10 Mb/s and 100 Mb/s Auto-Negotiation on twisted pair

## 28.1 Overview

### 28.1.1 Scope

Clause 28 describes the Auto-Negotiation function that allows a device to advertise enhanced modes of operation it possesses to a device at the remote end of a link segment and to detect corresponding enhanced operational modes that the other device may be advertising.

The objective of the Auto-Negotiation function is to provide the means to exchange information between two devices that share a link segment and to automatically configure both devices to take maximum advantage of their abilities. Auto-Negotiation is performed using a modified 10BASE-T link integrity test pulse sequence, such that no packet or upper layer protocol overhead is added to the network devices (see figure 28-1). Auto-Negotiation does not test the link segment characteristics (see 28.1.4).

The function allows the devices at both ends of a link segment to advertise abilities, acknowledge receipt and understanding of the common mode(s) of operation that both devices share, and to reject the use of operational modes that are not shared by both devices. Where more than one common mode exists between the two devices, a mechanism is provided to allow the devices to resolve to a single mode of operation using a predetermined priority resolution function. The Auto-Negotiation function allows the devices to switch between the various operational modes in an ordered fashion, permits management to disable or enable the Auto-Negotiation function, and allows management to select a specific operational mode. The Auto-Negotiation function also provides a Parallel Detection function to allow 10BASE-T, 100BASE-TX, and 100BASE-T4 compatible devices to be recognized, even though they may not provide Auto-Negotiation.

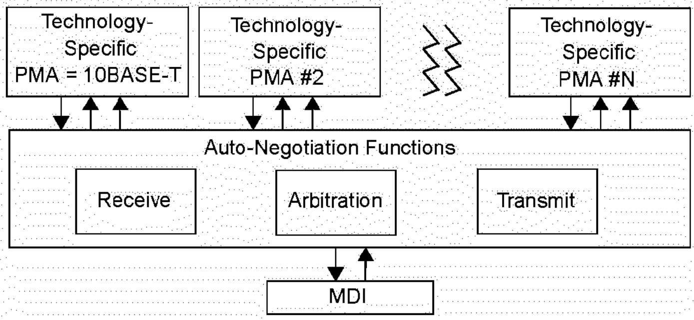


Figure 28-1—High-level model

The basic mechanism to achieve Auto-Negotiation is to pass information encapsulated within a burst of closely spaced link integrity test pulses that individually meet the 10BASE-T Transmitter Waveform for Link Test Pulse (figure 14-12). This burst of pulses is referred to as a Fast Link Pulse (FLP) Burst. Each device capable of Auto-Negotiation issues FLP Bursts at power up, on command from management, or due to user interaction. The FLP Burst consists of a series of link integrity test pulses that form an alternating clock/data sequence. Extraction of the data bits from the FLP Burst yields a Link Code Word that identifies the operational modes supported by the remote device, as well as some information used for the Auto-Negotiation function's handshake mechanism.

To maintain interoperability with existing 10BASE-T devices, the function also supports the reception of 10BASE-T compliant link integrity test pulses. 10BASE-T link pulse activity is referred to as the Normal Link Pulse (NLP) sequence and is defined in 14.2.1.1. A device that fails to respond to the FLP Burst sequence by returning only the NLP sequence is treated as a 10BASE-T compatible device.

### 28.1.2 Application perspective/objectives

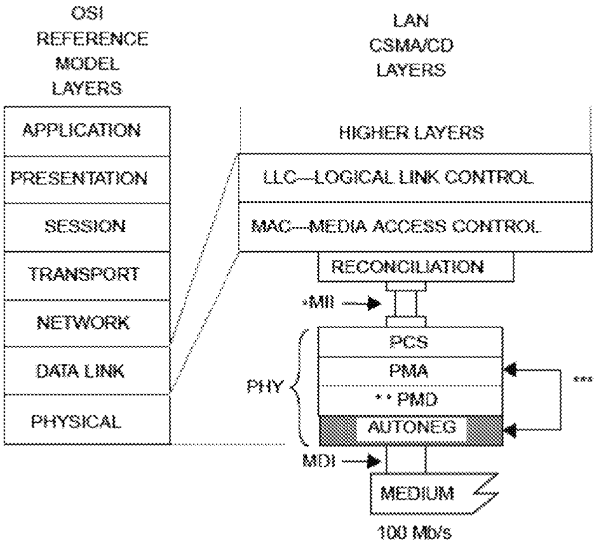
The Auto-Negotiation function is designed to be expandable and allow IEEE 802.3 compatible devices using an eight-pin modular connector to self-configure a jointly compatible operating mode. Implementation of the Auto-Negotiation function is optional. However, it is highly recommended that this method alone be utilized to perform the negotiation of the link operation.

The following are the objectives of Auto-Negotiation:

a) Must interoperate with the IEEE 802.3 10BASE-T installed base.
b) Must allow automatic upgrade from the 10BASE-T mode to the desired "High-Performance Mode."
c) Requires that the 10BASE-T data service is the Lowest Common Denominator (LCD) that can be resolved. A 10BASE-T PMA is not required to be implemented, however. Only the NLP Receive Link Integrity Test function is required.
d) Reasonable and cost-effective to implement.
e) Must provide a sufficiently extensible code space to
    1) Meet existing and future requirements.
    2) Allow simple extension without impacting the installed base.
    3) Accommodate remote fault signals.
    4) Accommodate link partner ability detection.
f) Must allow manual or Network Management configuration to override the Auto-Negotiation.
g) Must be capable of operation in the absence of Network Management.
h) Must not preclude the ability to negotiate "back" to the 10BASE-T operational mode.
i) Must operate when
    1) The link is initially electrically connected.
    2) A device at either end of the link is powered up, reset, or a renegotiation request is made.
j) The Auto-Negotiation function may be enabled by automatic, manual, or Network Management intervention.
k) Completes the base page Auto-Negotiation function in a bounded time period.
l) Will provide the basis for the link establishment process in future CSMA/CD compatible LAN standards that use an eight-pin modular connector.
m) Must not cause corruption of IEEE 802.3 Layer Management statistics.
n) Operates using a peer-to-peer exchange of information with no requirement for a master device (not master-slave).
o) Must be robust in the UTP cable noise environment.
p) Must not significantly impact EMI/RFI emissions.

### 28.1.3 Relationship to ISO/IEC 8802-3

The Auto-Negotiation function is provided at the Physical Layer of the OSI reference model as shown in figure 28-2. Devices that support multiple modes of operation may advertise this fact using this function. The actual transfer of information of ability is observable only at the MDI or on the medium. Auto-Negotiation signaling does not occur across either the AUI or MII. Control of the Auto-Negotiation function may be supported through the Management Interface of the MII or equivalent. If an explicit embodiment of the MII is supported, the control and status registers to support the Auto-Negotiation function shall be implemented in accordance with the definitions in clause 22 and 28.2.4. If a physical embodiment of the MII management is not present, then it is strongly recommended that the implementation provide control and status mechanisms equivalent to those described in clause 22 and 28.2.4 for manual and/or management interaction.



MDI = MEDIUM DEPENDENT INTERFACE
MII = MEDIA INDEPENDENT INTERFACE
AUTONEG = AUTO-NEGOTIATION

PCS = PHYSICAL CODING SUBLAYER
PMA = PHYSICAL MEDIUM ATTACHMENT
PHY = PHYSICAL LAYER DEVICE
PMD = PHYSICAL MEDIUM DEPENDENT

* MII is optional for 10 Mb/s DTEs and for 100 Mb/s systems and is not specified for 1 Mb/s systems.
** PMD is specified for 100BASE-X only; 100BASE-T4 does not use this layer.
*** AUTONEG communicates with the PMA sublayer through the PMA service interface messages PMA_LINK.request and PMA_LINK.indicate.

**Figure 28-2—Location of Auto-Negotiation function within the ISO reference model**

### 28.1.4 Compatibility considerations

The Auto-Negotiation function is designed to be completely backwards compatible and interoperable with 10BASE-T compliant devices. In order to achieve this, a device supporting the Auto-Negotiation function must provide the NLP Receive Link Integrity Test function as defined in figure 28-17. The Auto-Negotiation function also supports connection to 100BASE-TX and 100BASE-T4 devices without Auto-Negotiation through the Parallel Detection function. Connection to technologies other than 10BASE-T, 100BASE-TX, or 100BASE-T4 that do not incorporate Auto-Negotiation is not supported.

Implementation of the Auto-Negotiation function is optional. For CSMA/CD compatible devices that use the eight-pin modular connector of ISO/IEC 8877: 1992 and that also encompass multiple operational modes, if a signaling method is used to automatically configure the preferred mode of operation, then the Auto-Negotiation function shall be used in compliance with clause 28. If the device uses 10BASE-T compatible link signaling to advertise non-CSMA/CD abilities, the device shall implement the Auto-Negotiation function as administered by this specification. All future CSMA/CD implementations that use an eight-pin modular connector shall be interoperable with devices supporting clause 28. If the implementor of a non-CSMA/CD eight-pin modular device wishes to assure that its operation does not conflict with CSMA/CD devices, then adherence to clause 28 is recommended.

While this Auto-Negotiation function must be implemented in CSMA/CD compatible devices that utilize the eight-pin modular connector, encompass multiple operational modes, and offer an Auto-Negotiation mechanism, the use of this function does not mandate that the 10BASE-T packet data communication service must exist. A device that employs this function must support the 10BASE-T Link Integrity Test function through the NLP Receive Link Integrity Test state diagram. The device may also need to support other technology-dependent link test functions depending on the modes supported. Auto-Negotiation does not perform cable tests, such as detect number of conductor pairs (if more than two pairs are required) or cable performance measurements. Some PHYs that explicitly require use of high-performance cables, may require knowledge of the cable type, or additional robustness tests (such as monitoring CRC or framing errors) to determine if the link segment is adequate.

### 28.1.4.1 Interoperability with existing 10BASE-T devices

During Auto-Negotiation, FLP Bursts separated by 16 ± 8 ms are transmitted. The FLP Burst itself is a series of pulses separated by 62.5 ± 7 µs. The timing of FLP Bursts will cause a 10BASE-T device that is in the LINK TEST PASS state to remain in the LINK TEST PASS state while receiving FLP Bursts. An Auto-Negotiation able device must recognize the NLP sequence from a 10BASE-T Link Partner, cease transmission of FLP Bursts, and enable the 10BASE-T PMA, if present. If the NLP sequence is detected and if the Auto-Negotiation able device does not have a 10BASE-T PMA, it will cease transmission of FLP Bursts, forcing the 10BASE-T Link Partner into the LINK TEST FAIL state(s) as indicated in figure 14-6.

NOTE—Auto-Negotiation does not support the transmission of the NLP sequence. The 10BASE-T PMA provides this function if it is connected to the MDI. In the case where an Auto-Negotiation able device without a 10BASE-T PMA is connected to a 10BASE-T device without Auto-Negotiation, the NLP sequence is not transmitted because the Auto-Negotiation function has no 10BASE-T PMA to enable that can transmit the NLP sequence.

### 28.1.4.2 Interoperability with Auto-Negotiation compatible devices

An Auto-Negotiation compatible device decodes the base Link Code Word from the FLP Burst, and examines the contents for the highest common ability that both devices share. Both devices acknowledge correct receipt of each other's base Link Code Words by responding with FLP Bursts containing the Acknowledge Bit set. After both devices complete acknowledgment, and optionally, Next Page exchange, both devices enable the highest common mode negotiated. The highest common mode is resolved using the priority resolution hierarchy specified in annex 28B. It may subsequently be the responsibility of a technology-dependent link integrity test function to verify operation of the link prior to enabling the data service.

### 28.1.4.3 Cabling compatibility with Auto-Negotiation

Provision has been made within Auto-Negotiation to limit the resulting link configuration in situations where the cabling may not support the highest common capability of the two end points. The system administrator/installer must take the cabling capability into consideration when configuring a hub port's advertised capability. That is, the advertised capability of a hub port should not result in an operational mode that is not compatible with the cabling.

## 28.2 Functional specifications

The Auto-Negotiation function provides a mechanism to control connection of a single MDI to a single PMA type, where more than one PMA type may exist. Management may provide additional control of Auto-Negotiation through the Management function, but the presence of a management agent is not required.

The Auto-Negotiation function shall provide the Auto-Negotiation Transmit, Receive, Arbitration, and NLP Receive Link Integrity Test functions and comply with the state diagrams of figures 28-14 to 28-17. The Auto-Negotiation functions shall interact with the technology-dependent PMAs through the Technology-Dependent Interface. Technology-dependent PMAs include, but are not limited to, 100BASE-TX and 100BASE-T4. Technology-dependent link integrity test functions shall be implemented and interfaced to only if the device supports the given technology. For example, a 10BASE-T and 100BASE-TX Auto-Negotiation able device must implement and interface to the 100BASE-TX PMA/link integrity test function, but does not need to include the 100BASE-T4 PMA/Link Integrity Test function. The Auto-Negotiation function shall provide an optional Management function that provides a control and status mechanism.

### 28.2.1 Transmit function requirements

The Transmit function provides the ability to transmit FLP Bursts. The first FLP Bursts exchanged by the Local Device and its Link Partner after Power-On, link restart, or renegotiation contain the base Link Code Word defined in 28.2.1.2. The Local Device may modify the Link Code Word to disable an ability it possesses, but will not transmit an ability it does not possess. This makes possible the distinction between local abilities and advertised abilities so that multimode devices may Auto-Negotiate to a mode lower in priority than the highest common local ability.

### 28.2.1.1 Link pulse transmission

Auto-Negotiation's method of communication builds upon the link pulse mechanism employed by 10BASE-T MAUs to detect the status of the link. Compliant 10BASE-T MAUs transmit link integrity test pulses as a mechanism to determine if the link segment is operational in the absence of packet data. The 10BASE-T NLP sequence is a pulse (figure 14-12) transmitted every 16 ± 8 ms while the data transmitter is idle.

Auto-Negotiation substitutes the FLP Burst in place of the single 10BASE-T link integrity test pulse within the NLP sequence (figure 28-3). The FLP Burst encodes the data that is used to control the Auto-Negotiation function. FLP Bursts shall not be transmitted when Auto-Negotiation is complete and the highest common denominator PMA has been enabled.

FLP Bursts were designed to allow use beyond initial link Auto-Negotiation, such as for a link monitor type function. However, use of FLP Bursts beyond the current definition for link startup shall be prohibited. Definition of the use of FLP Bursts while in the FLP LINK GOOD state is reserved.

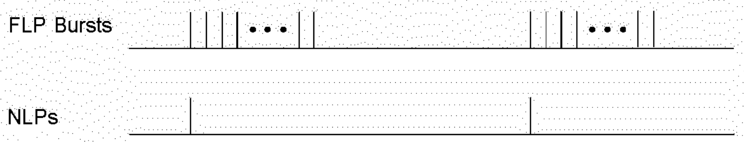


Figure 28-3—FLP Burst sequence to NLP sequence mapping

### 28.2.1.1.1 FLP burst encoding

FLP Bursts shall be composed of link pulses meeting the requirements of figure 14-12. A Fast Link Pulse Burst consists of 33 pulse positions. The 17 odd-numbered pulse positions shall contain a link pulse and represent clock information. The 16 even-numbered pulse positions shall represent data information as follows: a link pulse present in an even-numbered pulse position represents a logic one, and a link pulse absent from an even-numbered pulse position represents a logic zero. Clock pulses are differentiated from data pulses by the spacing between pulses as shown in figure 28-5 and enumerated in table 28-1.

The encoding of data using pulses in an FLP Burst is illustrated in figure 28-4.

### 28.2.1.1.2 Transmit timing

The first pulse in an FLP Burst shall be defined as a clock pulse. Clock pulses within an FLP Burst shall be spaced at 125 ± 14 µs. If the data bit representation of logic one is to be transmitted, a pulse shall occur 62.5 ± 7 µs after the preceding clock pulse. If a data bit representing logic zero is to be transmitted, there shall be no link integrity test pulses within 111 µs of the preceding clock pulse.

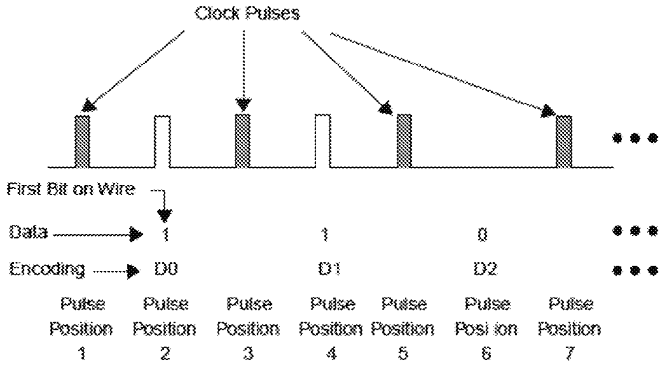


Figure 28-4—Data bit encoding within FLP Bursts

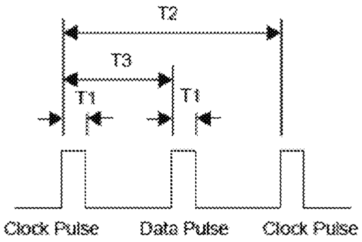


Figure 28-5—FLP Burst pulse-to-pulse timing

The first link pulse in consecutive FLP Bursts shall occur at a 16 ± 8 ms interval (figure 28-6).

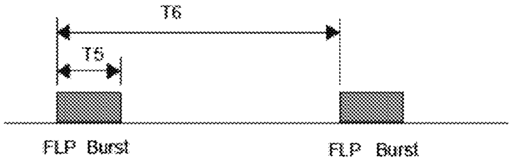


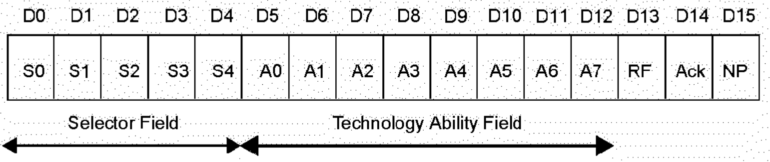
Figure 28-6—FLP Burst to FLP Burst timing

Table 28-1—FLP Burst timing summary

| # | Parameter | Min. | Typ. | Max. | Units |
|---|---|---|---|---|---|
| T1 | Clock/Data Pulse Width (figure 14-12) | | 100 | | ns |
| T2 | Clock Pulse to Clock Pulse | 111 | 125 | 139 | µs |
| T3 | Clock Pulse to Data Pulse (Data = 1) | 55.5 | 62.5 | 69.5 | µs |
| T4 | Pulses in a Burst | 17 | | 33 | # |
| T5 | Burst Width | | 2 | | ms |
| T6 | FLP Burst to FLP Burst | 8 | 16 | 24 | ms |

#### 28.2.1.2 Link Code Word encoding

The base Link Code Word (base page) transmitted within an FLP Burst shall convey the encoding shown in figure 28-7. The Auto-Negotiation function may support additional pages using the Next Page function. Encodings for the Link Code Word(s) used in Next Page exchange are defined in 28.2.3.4. In an FLP Burst, D0 shall be the first bit transmitted.



| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| S0 | S1 | S2 | S3 | S4 | A0 | A1 | A2 | A3 | A4 | A5 | A6 | A7 | RF | Ack | NP |

**Figure 28-7—Base page encoding**

##### 28.2.1.2.1 Selector Field

Selector Field (S[4:0]) is a five bit wide field, encoding 32 possible messages. Selector Field encoding definitions are shown in annex 28A. Combinations not specified are reserved for future use. Reserved combinations of the Selector Field shall not be transmitted.

##### 28.2.1.2.2 Technology Ability Field

Technology Ability Field (A[7:0]) is an eight bit wide field containing information indicating supported technologies specific to the selector field value. These bits are mapped to individual technologies such that abilities are advertised in parallel for a single selector field value. The Technology Ability Field encoding for the IEEE 802.3 selector is described in annex 28B.2. Multiple technologies may be advertised in the Link Code Word. A device shall support the data service ability for a technology it advertises. It is the responsibility of the Arbitration function to determine the common mode of operation shared by a Link Partner and to resolve multiple common modes.

NOTE—While devices using a Selector Field value other than the IEEE 802.3 Selector Field value are free to define the Technology Ability Field bits, it is recommended that the 10BASE-T bit be encoded in the same bit position as in the IEEE 802.3 selector. A common bit position can be important if the technology using the other selector will ever coexist on a device that also offers a 10BASE-T mode.

##### 28.2.1.2.3 Remote Fault

Remote Fault (RF) is encoded in bit D13 of the base Link Code Word. The default value is logic zero. The Remote Fault bit provides a standard transport mechanism for the transmission of simple fault information. When the RF bit in the Auto-Negotiation advertisement register (register 4) is set to logic one, the RF bit in the transmitted base Link Code Word is set to logic one. When the RF bit in the received base Link Code Word is set to logic one, the Remote Fault bit in the MII status register (register 1) will be set to logic one, if the MII management function is present.

The Remote Fault bit shall be used in accordance with the Remote Fault function specifications (28.2.3.5).

##### 28.2.1.2.4 Acknowledge

Acknowledge (Ack) is used by the Auto-Negotiation function to indicate that a device has successfully received its Link Partner's Link Code Word. The Acknowledge Bit is encoded in bit D14 regardless of the value of the Selector Field or Link Code Word encoding. If no Next Page information is to be sent, this bit shall be set to logic one in the Link Code Word after the reception of at least three consecutive and consistent FLP Bursts (ignoring the Acknowledge bit value). If Next Page information is to be sent, this bit shall be set to logic one after the device has successfully received at least three consecutive and matching FLP Bursts (ignoring the Acknowledge bit value), and will remain set until the Next Page information has been loaded into the Auto-Negotiation Next Page register (register 7). In order to save the current received Link Code Word, this must be read from the Auto-Negotiation link partner ability register (register 6) before the Next Page of transmit information is loaded into the Auto-Negotiation Next Page register. After the COMPLETE ACKNOWLEDGE state has been entered, the Link Code Word shall be transmitted six to eight (inclusive) times.

#### 28.2.1.2.5 Next Page

Next Page (NP) is encoded in bit D15 regardless of the Selector Field value or Link Code Word encoding. Support for transmission and reception of additional Link Code Word encodings is optional. If Next Page ability is not supported, the NP bit shall always be set to logic zero. If a device implements Next Page ability and wishes to engage in Next Page exchange, it shall set the NP bit to logic one. A device may implement Next Page ability and choose not to engage in Next Page exchange by setting the NP bit to a logic zero. The Next Page function is defined in 28.2.3.4.

### 28.2.1.3 Transmit Switch function

The Transmit Switch function shall enable the transmit path from a single technology-dependent PMA to the MDI once a highest common denominator choice has been made and Auto-Negotiation has completed.

During Auto-Negotiation, the Transmit Switch function shall connect only the FLP Burst generator controlled by the Transmit State Diagram, figure 28-14, to the MDI.

When a PMA is connected to the MDI through the Transmit Switch function, the signals at the MDI shall conform to all of the PHY's specifications.

### 28.2.2 Receive function requirements

The Receive function detects the NLP sequence using the NLP Receive Link Integrity Test function of figure 28-17. The NLP Receive Link Integrity Test function will not detect link pass based on carrier sense.

The Receive function detects the FLP Burst sequence, decodes the information contained within, and stores the data in rx_link_code_word[16:1]. The Receive function incorporates a receive switch to control connection to the 100BASE-TX or 100BASE-T4 PMAs in addition to the NLP Receive Link Integrity Test function, excluding the 10BASE-T Link Integrity Test function present in a 10BASE-T PMA. If Auto-Negotiation detects link_status=READY from any of the technology-dependent PMAs prior to FLP Burst detection, the autoneg_wait_timer (28.3.2) is started. If any other technology-dependent PMA indicates link_status=READY when the autoneg_wait_timer expires, Auto-Negotiation will not allow any data service to be enabled and may signal this as a remote fault to the Link Partner using the base page and will flag this in the Local Device by setting the Parallel Detection Fault bit (6.4) in the Auto-Negotiation expansion register. If a 10BASE-T PMA exists above the Auto-Negotiation function, it is not permitted to receive MDI activity in parallel with the NLP Receive Link Integrity Test function or any other technology-dependent function.

#### 28.2.2.1 FLP Burst ability detection and decoding

In figures 28-8 to 28-10, the symbol "$t_0$=0" indicates the event that caused the timers described to start, and all subsequent times given are referenced from that point. All timers referenced shall expire within the range specified in table 28-8 in 28.3.2.

The Receive function shall identify the Link Partner as Auto-Negotiation able if it receives 6 to 17 (inclusive) consecutive link pulses that are separated by at least flp_test_min_timer time (5–25 µs) but less than flp_test_max_timer time (165–185 µs) as shown in figure 28-8. The information contained in the FLP Burst that identifies the Link Partner as Auto-Negotiation able shall not be passed to the Arbitration function if the FLP Burst is not complete. The Receive function may use the FLP Burst that identifies the Link Partner as Auto-Negotiation able for ability matching if the FLP Burst is complete. However, it is not required to use this FLP Burst for any purpose other than identification of the Link Partner as Auto-Negotiation able. Implementations may ignore multiple FLP Bursts before identifying the Link Partner as Auto-Negotiation able to allow for potential receive equalization time.

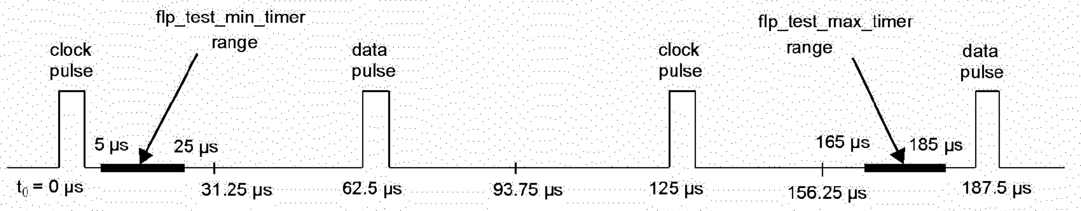


**Figure 28-8—FLP detect timers (flp_test_min/max_timers)**

The Receive function captures and decodes link pulses received in FLP Bursts. The first link pulse in an FLP Burst shall be interpreted as a clock link pulse. Detection of a clock link pulse shall restart the data_detect_min_timer and data_detect_max timer. The data_detect_min/max_timers enable the receiver to distinguish data pulses from clock pulses and logic one data from logic zero data, as follows:

a) If, during an FLP Burst, a link pulse is received when the data_detect_min_timer has expired while the data_detect_max_timer has not expired, the data bit shall be interpreted as a logic one (figure 28-9).
b) If, during an FLP Burst, a link pulse is received after the data_detect_max_timer has expired, the data bit shall be interpreted as a logic zero (figure 28-9) and that link pulse shall be interpreted as a clock link pulse.

As each data bit is identified it is stored in the appropriate rx_link_code_word[16:1] element.

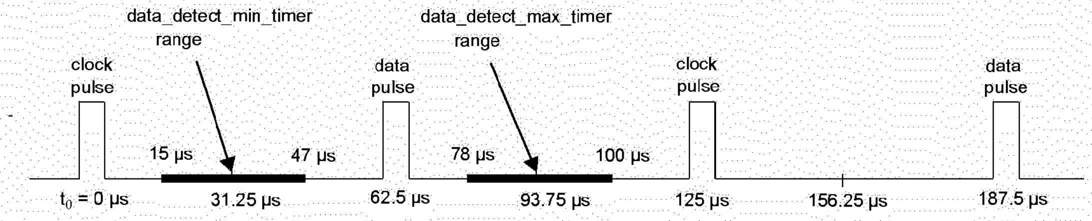
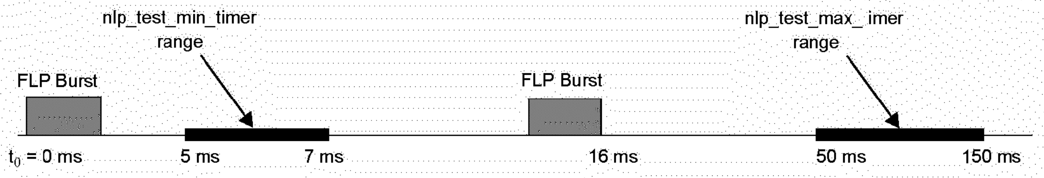


**Figure 28-9—FLP data detect timers (data_detect_min/max_timers)**

FLP Bursts conforming to the nlp_test_min_timer and nlp_test_max_timer timing as shown in figure 28-10 shall be considered to have valid separation.

### 28.2.2.2 NLP detection

NLP detection is accomplished via the NLP Receive Link Integrity Test function in figure 28-17. The NLP Receive Link Integrity Test function is a modification of the original 10BASE-T Link Integrity Test function (figure 14-6), where the detection of receive activity will not cause a transition to the LINK TEST PASS state during Auto-Negotiation. The NLP Receive Link Integrity Test function also incorporates the Technology-Dependent Interface requirements.



NOTE—The reference for the starting of the nlp_test_min_timer is from the beginning of the FLP Burst, as shown by $t_0$, while the reference for the starting of the nlp_test_max_timer is from the expiration of the nlp_test_min_timer.

**Figure 28-10—FLP Burst timer (nlp_test_min/max_timers)**

#### 28.2.2.3 Receive Switch function

The Receive Switch function shall enable the receive path from the MDI to a single technology-dependent PMA once a highest common denominator choice has been made and Auto-Negotiation has completed.

During Auto-Negotiation, the Receive Switch function shall connect both the FLP Burst receiver controlled by the Receive state diagram, figure 28-15, and the NLP Receive Link Integrity Test state diagram, figure 28-17, to the MDI. During Auto-Negotiation, the Receive Switch function shall also connect the 100BASE-TX and 100BASE-T4 PMA receivers to the MDI if the 100BASE-TX and/or 100BASE-T4 PMAs are present.

When a PMA is connected to the MDI through the Receive Switch function, the signals at the PMA shall conform to all of the PHY's specifications.

#### 28.2.2.4 Link Code Word matching

The Receive function shall generate ability_match, acknowledge_match, and consistency_match variables as defined in 28.3.1.

### 28.2.3 Arbitration function requirements

The Arbitration function ensures proper sequencing of the Auto-Negotiation function using the Transmit function and Receive function. The Arbitration function enables the Transmit function to advertise and acknowledge abilities. Upon indication of acknowledgment, the Arbitration function determines the highest common denominator using the priority resolution function and enables the appropriate technology-dependent PMA via the Technology-Dependent Interface (28.2.6).

#### 28.2.3.1 Parallel detection function

The Local Device detects a Link Partner that supports Auto-Negotiation by FLP Burst detection. The Parallel Detection function allows detection of Link Partners that support 100BASE-TX, 100BASE-T4, and/or 10BASE-T, but do not support Auto-Negotiation. Prior to detection of FLP Bursts, the Receive Switch shall direct MDI receive activity to the NLP Receive Link Integrity Test state diagram, 100BASE-TX and 100BASE-T4 PMAs, if present, but shall not direct MDI receive activity to the 10BASE-T or any other PMA. If at least one of the 100BASE-TX, 100BASE-T4, or NLP Receive Link Integrity Test functions establishes link_status=READY, the LINK STATUS CHECK state is entered and the autoneg_wait_timer is started. If exactly one link_status=READY indication is present when the autoneg_wait_timer expires, then Auto-Negotiation shall set link_control=ENABLE for the PMA indicating link_status=READY. If a PMA is enabled, the Arbitration function shall set link_control=DISABLE to all other PMAs and indicate that Auto-Negotiation has completed. On transition to the FLP LINK GOOD CHECK state from the LINK STA-

TUS CHECK state the Parallel Detection function shall set the bit in the link partner ability register (register 5) corresponding to the technology detected by the Parallel Detection function.

NOTES

1—Native 10BASE-T devices will be detected by the NLP Receive Link Integrity Test function, an integrated part of the Auto-Negotiation function. Hence, Parallel Detection for the 10BASE-T PMA is not required or allowed.

2—When selecting the highest common denominator through the Parallel Detection function, only the half-duplex mode corresponding to the selected PMA may automatically be detected.

#### 28.2.3.2 Renegotiation function

A renegotiation request from any entity, such as a management agent, shall cause the Arbitration function to disable all technology-dependent PMAs and halt any transmit data and link pulse activity until the break_link_timer expires (28.3.2). Consequently, the Link Partner will go into link fail and normal Auto-Negotiation resumes. The Local Device shall resume Auto-Negotiation after the break_link_timer has expired by issuing FLP Bursts with the base page valid in tx_link_code_word[16:1].

Once Auto-Negotiation has completed, renegotiation will take place if the Highest Common Denominator technology that receives link_control=ENABLE returns link_status=FAIL. To allow the PMA an opportunity to determine link integrity using its own link integrity test function, the link_fail_inhibit_timer qualifies the link_status=FAIL indication such that renegotiation takes place if the link_fail_inhibit_timer has expired and the PMA still indicates link_status=FAIL or link_status=READY.

#### 28.2.3.3 Priority Resolution function

Since a Local Device and a Link Partner may have multiple common abilities, a mechanism to resolve which mode to configure is required. The mechanism used by Auto-Negotiation is a Priority Resolution function that predefines the hierarchy of supported technologies. The single PMA enabled to connect to the MDI by Auto-Negotiation shall be the technology corresponding to the bit in the Technology Ability Field common to the Local Device and Link Partner that has the highest priority as defined in annex 28B. This technology is referred to as the Highest Common Denominator, or HCD, technology. If the Local Device receives a Technology Ability Field with a bit set that is reserved, the Local Device shall ignore that bit for priority resolution. Determination of the HCD technology occurs on entrance to the FLP LINK GOOD CHECK state. In the event that a technology is chosen through the Parallel Detection function, that technology shall be considered the highest common denominator (HCD) technology. In the event that there is no common technology, HCD shall have a value of "NULL," indicating that no PMA receives link_control=ENABLE, and link_status_[HCD]=FAIL.

#### 28.2.3.4 Next Page function

The Next Page function uses the standard Auto-Negotiation arbitration mechanisms to allow exchange of arbitrary pieces of data. Data is carried by optional Next Pages of information, which follow the transmission and acknowledgment procedures used for the base Link Code Word. Two types of Next Page encodings are defined: Message Pages and Unformatted Pages.

A dual acknowledgment system is used. Acknowledge (Ack) is used to acknowledge receipt of the information; Acknowledge 2 (Ack2) is used to indicate that the receiver is able to act on the information (or perform the task) defined in the message.

Next Page operation is controlled by the same two mandatory control bits, Next Page and Acknowledge, used in the Base Link Code Word. Setting the NP bit in the Base Link Code Word to logic one indicates that the device is Next Page Able. If both a device and its Link Partner are Next Page Able, then Next Page exchange may occur. If one or both devices are not Next Page Able, then Next Page exchange will not occur and, after the base Link Code Words have been exchanged, the FLP LINK GOOD CHECK state will be entered. The Toggle bit is used to ensure proper synchronization between the Local Device and the Link Partner.

Next Page exchange occurs after the base Link Code Words have been exchanged. Next Page exchange consists of using the normal Auto-Negotiation arbitration process to send Next Page messages. Two message encodings are defined: Message Pages, which contain predefined 11 bit codes, and Unformatted Pages. Unformatted Pages can be combined to send extended messages. If the Selector Field values do not match, then each series of Unformatted Pages shall be preceded by a Message Page containing a message code that defines how the following Unformatted Pages will be interpreted. If the Selector Field values match, then the convention governing the use of Message Pages shall be as defined by the Selector Field value definition. Any number of Next Pages may be sent in any order; however, it is recommended that the total number of Next Pages sent be kept small to minimize the link startup time.

Next Page transmission ends when both ends of a link segment set their Next Page bits to logic zero, indicating that neither has anything additional to transmit. It is possible for one device to have more pages to transmit than the other device. Once a device has completed transmission of its Next Page information, it shall transmit Message Pages with Null message codes and the NP bit set to logic zero while its Link Partner continues to transmit valid Next Pages. An Auto-Negotiation able device shall recognize reception of Message Pages with Null message codes as the end of its Link Partner's Next Page information.

#### 28.2.3.4.1 Next Page encodings

The Next Page shall use the encoding shown in figures 28-11 and 28-12 for the NP, Ack, MP, Ack2, and T bits. The 11-bit field D10–D0 shall be encoded as a Message Code Field if the MP bit is logic one and an Unformatted Code Field if MP is set to logic zero.

| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| M0 | M1 | M2 | M3 | M4 | M5 | M6 | M7 | M8 | M9 | M10 | T | Ack2 | MP | Ack | NP |

Message Code Field

**Figure 28-11—Message Page encoding**

| D0 | D1 | D2 | D3 | D4 | D5 | D6 | D7 | D8 | D9 | D10 | D11 | D12 | D13 | D14 | D15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| U0 | U1 | U2 | U3 | U4 | U5 | U6 | U7 | U8 | U9 | U10 | T | Ack2 | MP | Ack | NP |

Unformatted Code Field

**Figure 28-12—Unformatted Page encoding**

#### 28.2.3.4.2 Next Page

Next Page (NP) is used by the Next Page function to indicate whether or not this is the last Next Page to be transmitted. NP shall be set as follows:

logic zero = last page.
logic one = additional Next Page(s) will follow.

#### 28.2.3.4.3 Acknowledge

As defined in 28.2.1.2.4.

#### 28.2.3.4.4 Message Page

Message Page (MP) is used by the Next Page function to differentiate a Message Page from an Unformatted Page. MP shall be set as follows:

logic zero = Unformatted Page.
logic one = Message Page.

#### 28.2.3.4.5 Acknowledge 2

Acknowledge 2 (Ack2) is used by the Next Page function to indicate that a device has the ability to comply with the message. Ack2 shall be set as follows:

logic zero = cannot comply with message.
logic one = will comply with message.

#### 28.2.3.4.6 Toggle

Toggle (T) is used by the Arbitration function to ensure synchronization with the Link Partner during Next Page exchange. This bit shall always take the opposite value of the Toggle bit in the previously exchanged Link Code Word. The initial value of the Toggle bit in the first Next Page transmitted is the inverse of bit 11 in the base Link Code Word and, therefore, may assume a value of logic one or zero. The Toggle bit shall be set as follows:

logic zero = previous value of the transmitted Link Code Word equalled logic one.
logic one = previous value of the transmitted Link Code Word equalled logic zero.

#### 28.2.3.4.7 Message Page encoding

Message Pages are formatted pages that carry a single predefined Message Code, which is enumerated in annex 28C. Two-thousand and forty-eight Message Codes are available. The allocation of these codes will be controlled by the contents of annex 28C. If the Message Page bit is set to logic one, then the bit encoding of the Link Code Word shall be interpreted as a Message Page.

#### 28.2.3.4.8 Message Code Field

Message Code Field (M[10:0]) is an eleven bit wide field, encoding 2048 possible messages. Message Code Field definitions are shown in annex 28C. Combinations not specified are reserved for future use. Reserved combinations of the Message Code Field shall not be transmitted.

#### 28.2.3.4.9 Unformatted Page encoding

Unformatted Pages carry the messages indicated by Message Pages. Five control bits are predefined, the remaining 11 bits may take on an arbitrary value. If the Message Page bit is set to logic zero, then the bit encoding of the Link Code Word shall be interpreted as an Unformatted Page.

#### 28.2.3.4.10 Unformatted Code Field

Unformatted Code Field (U[10:0]) is an eleven bit wide field, which may contain an arbitrary value.

#### 28.2.3.4.11 Use of Next Pages

a) Both devices must indicate Next Page ability for either to commence exchange of Next Pages.
b) If both devices are Next Page able, then both devices shall send at least one Next Page.
c) Next Page exchange shall continue until neither device on a link has more pages to transmit as indicated by the NP bit. A Message Page with a Null Message Code Field value shall be sent if the device has no other information to transmit.
d) A Message Code can carry either a specific message or information that defines how following Unformatted Page(s) should be interpreted.
e) If a Message Code references Unformatted Pages, the Unformatted Pages shall immediately follow the referencing Message Code in the order specified by the Message Code.
f) Unformatted Page users are responsible for controlling the format and sequencing for their Unformatted Pages.

#### 28.2.3.4.12 MII register requirements

The Next Page Transmit register defined in 28.2.4.1.6 shall hold the Next Page to be sent by Auto-Negotiation. Received Next Pages may be stored in the Auto-Negotiation link partner ability register.

### 28.2.3.5 Remote fault sensing function

The Remote Fault function may indicate to the Link Partner that a fault condition has occurred using the Remote Fault bit and, optionally, the Next Page function.

Sensing of faults in a device as well as subsequent association of faults with the Remote Fault bit shall be optional. If the Local Device has no mechanism to detect a fault or associate a fault condition with the received Remote Fault bit indication, then it shall transmit the Remote Fault bit with the value contained in the Auto-Negotiation advertisement register bit (4.13).

A Local Device may indicate it has sensed a fault to its Link Partner by setting the Remote Fault bit in the Auto-Negotiation advertisement register and renegotiating.

If the Local Device sets the Remote Fault bit to logic one, it may also use the Next Page function to specify information about the fault that has occurred. Remote Fault Message Page Codes have been specified for this purpose.

The Remote Fault bit shall remain set until after successful negotiation with the base Link Code Word, at which time the Remote Fault bit shall be reset to a logic zero. On receipt of a base Link Code Word with the Remote Fault bit set to logic one, the device shall set the Remote Fault bit in the MII status register (1.4) to logic one if the MII management function is present.

## 28.2.4 Management function requirements

The management interface is used to communicate Auto-Negotiation information to the management entity. If an MII is physically implemented, then management access is via the MII Management interface. Where no physical embodiment of the MII exists, an equivalent to MII registers 0, 1, 4, 5, 6, and 7 (clause 22) are recommended to be provided.

### 28.2.4.1 Media Independent Interface

The Auto-Negotiation function shall have five dedicated registers:

a) MII control register (register 0).
b) MII status register (register 1).

c) Auto-Negotiation advertisement register (register 4).
d) Auto-Negotiation link partner ability register (register 5).
e) Auto-Negotiation expansion register (register 6).

If the Next Page function is implemented, the Auto-Negotiation Next Page Transmit Register (register 7) shall be implemented.

#### 28.2.4.1.1 MII control register

MII control register (register 0) provides the mechanism to disable/enable and/or restart Auto-Negotiation. The definition for this register is provided in 22.2.4.1.

The Auto-Negotiation function shall be enabled by setting bit 0.12 to a logic one. If bit 0.12 is set to a logic one, then bits 0.13 and 0.8 shall have no effect on the link configuration, and the Auto-Negotiation process will determine the link configuration. If bit 0.12 is cleared to logic zero, then bits 0.13 and 0.8 will determine the link configuration regardless of the prior state of the link configuration and the Auto-Negotiation process.

A PHY shall return a value of one in bit 0.9 until the Auto-Negotiation process has been initiated. The Auto-Negotiation process shall be initiated by setting bit 0.9 to a logic one. If Auto-Negotiation was completed prior to this bit being set, the process shall be reinitiated. If a PHY reports via bit 1.3 that it lacks the ability to perform Auto-Negotiation, then this bit will have no meaning, and should be written as zero. This bit is self-clearing. The Auto-Negotiation process shall not be affected by clearing this bit to logic zero.

#### 28.2.4.1.2 MII status register

The MII status register (register 1) includes information about all modes of operations supported by the Local Device's PHY, the status of Auto-Negotiation, and whether the Auto-Negotiation function is supported by the PHY or not. The definition for this register is provided in 22.2.4.2.

When read as a logic one, bit 1.5 indicates that the Auto-Negotiation process has been completed, and that the contents of registers 4, 5, and 6 are valid. When read as a logic zero, bit 1.5 indicates that the Auto-Negotiation process has not been completed, and that the contents of registers 4, 5, and 6 are meaningless. A PHY shall return a value of zero in bit 1.5 if Auto-Negotiation is disabled by clearing bit 0.12. A PHY shall also return a value of zero in bit 1.5 if it lacks the ability to perform Auto-Negotiation.

When read as logic one, bit 1.4 indicates that a remote fault condition has been detected. The type of fault as well as the criteria and method of fault detection is PHY specific. The Remote Fault bit shall be implemented with a latching function, such that the occurrence of a remote fault will cause the Remote Fault bit to become set and remain set until it is cleared. The Remote Fault bit shall be cleared each time register 1 is read via the management interface, and shall also be cleared by a PHY reset.

When read as a one, bit 1.3 indicates that the PHY has the ability to perform Auto-Negotiation. When read as a logic zero, bit 1.3 indicates that the PHY lacks the ability to perform Auto-Negotiation.

#### 28.2.4.1.3 Auto-Negotiation advertisement register (register 4) (R/W)

This register contains the Advertised Ability of the PHY. (See table 28-2). The bit definition for the base page is defined in 28.2.1.2. On power-up, before Auto-Negotiation starts, this register shall have the following configuration: The Selector Field (4.4:0) is set to an appropriate code as specified in annex 28A. The Acknowledge bit (4.14) is set to logic zero. The Technology Ability Field (4.12:5) is set based on the values set in the MII status register (register 1) (1.15:11) or equivalent.

Only the bits in the Technology Ability Field that represent the technologies supported by the Local Device may be set. Any of the Technology Ability Field bits that may be set can also be cleared by management

**Table 28-2—Advertisement register bit definitions**

| Bit(s) | Name | Description | R/W |
|---|---|---|---|
| 4.15 | Next Page | See 28.2.1.2 | R/W |
| 4.14 | Reserved | Write as zero, ignore on read | RO |
| 4.13 | Remote Fault | See 28.2.1.2 | R/W |
| 4.12:5 | Technology Ability Field | See 28.2.1.2 | R/W |
| 4.4:0 | Selector Field | See 28.2.1.2 | R/W |

before a renegotiation. This can be used to enable management to Auto-Negotiate to an alternate common mode.

The management entity may initiate renegotiation with the Link Partner using alternate abilities by setting the Selector Field (4.4:0) and Technology Ability Field (4.12:5) to indicate the preferred mode of operation and setting the Restart Auto-Negotiation bit (0.9) in the control register (register 0) to logic one.

Any writes to this register prior to completion of Auto-Negotiation as indicated by bit 1.5 should be followed by a renegotiation for the new values to be properly used for Auto-Negotiation. Once Auto-Negotiation has completed, this register value may be examined by software to determine the highest common denominator technology.

### 28.2.4.1.4 Auto-Negotiation link partner ability register (register 5) (RO)

All of the bits in the Auto-Negotiation link partner ability register are read only. A write to the Auto-Negotiation link partner ability register shall have no effect.

This register contains the Advertised Ability of the Link Partner's PHY. (See tables 28-3 and 28-4.) The bit definitions shall be a direct representation of the received Link Code Word (figure 28-7). Upon successful completion of Auto-Negotiation, status register (register 1) Auto-Negotiation Complete bit (1.5) shall be set to logic one. If the Next Page function is supported, the Auto-Negotiation link partner ability register may be used to store Link Partner Next Pages.

**Table 28-3—Link partner ability register bit definitions (Base Page)**

| Bit(s) | Name | Description | R/W |
|---|---|---|---|
| 5.15 | Next Page | See 28.2.1.2 | RO |
| 5.14 | Acknowledge | See 28.2.1.2 | RO |
| 5.13 | Remote Fault | See 28.2.1.2 | RO |
| 5.12:5 | Technology Ability Field | See 28.2.1.2 | RO |
| 5.4:0 | Selector Field | See 28.2.1.2 | RO |

The values contained in this register are only guaranteed to be valid once Auto-Negotiation has successfully completed, as indicated by bit 1.5 or, if used with Next Page exchange, after the Page Received bit (6.1) has been set to logic one.

### Table 28-4—Link partner ability register bit definitions (Next Page)

| Bit(s) | Name | Description | R/W |
|---|---|---|---|
| 5.15 | Next Page | See 28.2.3.4 | RO |
| 5.14 | Acknowledge | See 28.2.3.4 | RO |
| 5.13 | Message Page | See 28.2.3.4 | RO |
| 5.12 | Acknowledge 2 | See 28.2.3.4 | RO |
| 5.11 | Toggle | See 28.2.3.4 | RO |
| 5.10:0 | Message/Unformatted Code Field | See 28.2.3.4 | RO |

NOTE—If this register is used to store Link Partner Next Pages, the previous value of this register is assumed to be stored by a management entity that needs the information overwritten by subsequent Link Partner Next Pages.

#### 28.2.4.1.5 Auto-Negotiation expansion register (register 6) (RO)

All of the bits in the Auto-Negotiation expansion register are read only; a write to the Auto-Negotiation expansion register shall have no effect. (See table 28-5.)

### Table 28-5—Expansion register bit definitions

| Bit(s) | Name | Description | R/W | Default |
|---|---|---|---|---|
| 6.15:5 | Reserved | Write as zero, ignore on read | RO | 0 |
| 6.4 | Parallel Detection Fault | 1 = A fault has been detected via the Parallel Detection function.<br>0 = A fault has not been detected via the Parallel Detection function. | RO/ LH | 0 |
| 6.3 | Link Partner Next Page Able | 1 = Link Partner is Next Page able<br>0 = Link Partner is not Next Page able | RO | 0 |
| 6.2 | Next Page Able | 1 = Local Device is Next Page able<br>0 = Local Device is not Next Page able | RO | 0 |
| 6.1 | Page Received | 1 = A New Page has been received<br>0 = A New Page has not been received | RO/ LH | 0 |
| 6.0 | Link Partner Auto-Negotiation Able | 1 = Link Partner is Auto-Negotiation able<br>0 = Link Partner is not Auto-Negotiation able | RO | 0 |

Bits 6.15:5 are reserved for future Auto-Negotiation expansion.

The Parallel Detection Fault bit (6.4) shall be set to logic one to indicate that zero or more than one of the NLP Receive Link Integrity Test function, 100BASE-TX, or 100BASE-T4 PMAs have indicated link_status=READY when the autoneg_wait_timer expires. The Parallel Detection Fault bit shall be reset to logic zero on a read of the Auto-Negotiation expansion register (register 6).

The Link Partner Next Page Able bit (6.3) shall be set to logic one to indicate that the Link Partner supports the Next Page function. This bit shall be reset to logic zero to indicate that the Link Partner does not support the Next Page function.

The Next Page Able bit (6.2) shall be set to logic one to indicate that the Local Device supports the Next Page function. The Next Page Able bit (6.2) shall be set to logic zero if the Next Page function is not supported.

The Page Received bit (6.1) shall be set to logic one to indicate that a new Link Code Word has been received and stored in the Auto-Negotiation link partner ability register. The Page Received bit shall be reset to logic zero on a read of the Auto-Negotiation expansion register (register 6).

The Link Partner Auto-Negotiation Able bit (6.0) shall be set to logic one to indicate that the Link Partner is able to participate in the Auto-Negotiation function. This bit shall be reset to logic zero if the Link Partner is not Auto-Negotiation able.

#### 28.2.4.1.6 Auto-Negotiation Next Page transmit register (register 7) (R/W)

The Auto-Negotiation Next Page Transmit register contains the Next Page Link Code Word to be transmitted when Next Page ability is supported. (See table 28-6.) The contents are defined in 28.2.3.4. On power-up, this register shall contain the default value of 2001H, which represents a Message Page with the Message Code set to Null Message. This value may be replaced by any valid Next Page Message Code that the device wishes to transmit. Writing to this register shall set mr_next_page_loaded to true.

**Table 28-6—Next Page transmit register bit definitions**

| Bit(s) | Name | Description | R/W |
|---|---|---|---|
| 7.15 | Next Page | See 28.2.3.4 | R/W |
| 7.14 | Reserved | Write as 0, ignore on read | RO |
| 7.13 | Message Page | See 28.2.3.4 | R/W |
| 7.12 | Acknowledge 2 | See 28.2.3.4 | R/W |
| 7.11 | Toggle | See 28.2.3.4 | RO |
| 7.10:0 | Message/Unformatted Code Field | See 28.2.3.4 | R/W |

#### 28.2.4.1.7 State diagram variable to MII register mapping

The state diagrams of figures 28-14 to 28-17 generate and accept variables of the form “mr_x”, where x is an individual signal name. These variables comprise a management interface that may be connected to the MII management function or other equivalent function. Table 28-7 describes how the MII registers map to the management function interface signals.

#### 28.2.4.2 Auto-Negotiation managed object class

The Auto-Negotiation Managed Object Class is defined in clause 30.

### Table 28-7—State diagram variable to MII register mapping

| State diagram variable | MII register |
|---|---|
| mr_adv_ability[16:1] | 4.15:0 Auto-Negotiation advertisement register |
| mr_autoneg_complete | 1.5 Auto-Negotiation Complete |
| mr_autoneg_enable | 0.12 Auto-Negotiation Enable |
| mr_lp_adv_ability[16:1] | 5.15:0 Auto-Negotiation link partner ability register |
| mr_lp_autoneg_able | 6.0 Link Partner Auto-Negotiation Able |
| mr_lp_np_able | 6.3 Link Partner Next Page Able |
| mr_main_reset | 0.15 Reset |
| mr_next_page_loaded | Set on write to Auto-Negotiation Next Page Transmit register;<br>cleared by Arbitration state diagram |
| mr_np_able | 6.2 Next Page Able |
| mr_np_tx[16:1] | 7.15:0 Auto-Negotiation Next Page Transmit Register |
| mr_page_rx | 6.1 Page Received |
| mr_parallel_detection_fault | 6.4 Parallel Detection Fault |
| mr_restart_negotiation | 0.9 Auto-Negotiation Restart |
| set if Auto-Negotiation is available | 1.3 Auto-Negotiation Ability |

### 28.2.5 Absence of management function

In the absence of any management function, the advertised abilities shall be provided through a logical equivalent of mr_adv_ability[16:1]. A device shall comply with all Next Page function requirements, including the provision of the mr_np_able, mr_lp_np_able, and mr_next_page_loaded variables (or their logical equivalents), in order to permit the NP bit to be set to logic one in the transmitted Link Code Word.

NOTE—Storage of a valid base Link Code Word is required to prevent a deadlock situation where negotiation must start again while Next Pages are being transmitted. If a shared transmit register were used, then renegotiation could not occur when Next Pages were being transmitted because the base Link Code Word would not be available. This requirement can be met using a number of different implementations, including use of temporary registers or register stacks.

### 28.2.6 Technology-Dependent Interface

The Technology-Dependent Interface is the communication mechanism between each technology's PMA and the Auto-Negotiation function. Auto-Negotiation can support multiple technologies, all of which need not be implemented in a given device. Each of these technologies may utilize its own technology-dependent link integrity test function.

### 28.2.6.1 PMA_LINK.indicate

This primitive is generated by the PMA to indicate the status of the underlying medium. The purpose of this primitive is to give the PCS, repeater client, or Auto-Negotiation function a means of determining the validity of received code elements.

### 28.2.6.1.1 Semantics of the service primitive

PMA_LINK.indicate(link_status)

The link_status parameter shall assume one of three values: READY, OK, or FAIL, indicating whether the underlying receive channel is intact and ready to be enabled (READY), intact and enabled (OK), or not intact (FAIL). When link_status=FAIL or link_status=READY, the PMA_CARRIER.indicate and PMA_UNITDATA.indicate primitives are undefined.

### 28.2.6.1.2 When generated

A technology-dependent PMA and the NLP Receive Link Integrity Test state diagram (figure 28-17) shall generate this primitive to indicate the value of link_status.

### 28.2.6.1.3 Effect of receipt

The effect of receipt of this primitive shall be governed by the state diagrams of figure 28-16.

### 28.2.6.2 PMA_LINK.request

This primitive is generated by Auto-Negotiation to allow it to enable and disable operation of the PMA.

### 28.2.6.2.1 Semantics of the service primitive

PMA_LINK request(link_control)

The link_control parameter shall assume one of three values: SCAN_FOR_CARRIER, DISABLE, or ENABLE.

The link_control=SCAN_FOR_CARRIER mode is used by the Auto-Negotiation function prior to receiving any FLP Bursts or link_status=READY indications. During this mode, the PMA shall search for carrier and report link_status=READY when carrier is received, but no other actions shall be enabled.

The link_control=DISABLE mode shall be used by the Auto-Negotiation function to disable PMA processing.

The link_control=ENABLE mode shall be used by Auto-Negotiation to turn control over to a single PMA for all normal processing functions.

### 28.2.6.2.2 When generated

The Auto-Negotiation function shall generate this primitive to indicate to the PHY how to respond, in accordance with the state diagrams of figures 28-15 and 28-16.

Upon power-on or reset, if the Auto-Negotiation function is enabled (mr_autoneg_enable=true) the PMA_LINK request(DISABLE) message shall be issued to all technology-dependent PMAs. If Auto-Negotiation is disabled at any time including at power-on or reset, the state of PMA_LINK.request(link_control) is implementation dependent.

### 28.2.6.2.3 Effect of receipt

The effect of receipt of this primitive shall be governed by the NLP Receive Link Integrity Test state diagram (figure 28-17) and the receiving technology-dependent link integrity test function, based on the intent specified in the primitive semantics.

## 28.3 State diagrams and variable definitions

The notation used in the state diagrams (figures 28-14 to 28-17) follows the conventions in 21.5. State diagram variables follow the conventions of 21.5.2 except when the variable has a default value. Variables in a state diagram with default values evaluate to the variable default in each state where the variable value is not explicitly set. Variables using the "mr_x" notation do not have state diagram defaults; however, their appropriate initialization conditions when mapped to the MII interface are covered in 28.2.4 and 22.2.4. The variables, timers, and counters used in the state diagrams are defined in 28.3, 14.2.3, and 28.2.6.

Auto-Negotiation shall implement the Transmit state diagram, Receive state diagram, Arbitration state diagram, and NLP Receive Link Integrity Test state diagram as depicted in 28.3. Additional requirements to these state diagrams are made in the respective functional requirements sections. Options to these state diagrams clearly stated as such in the functional requirements sections or state diagrams shall be allowed. In the case of any ambiguity between stated requirements and the state diagrams, the state diagrams shall take precedence.

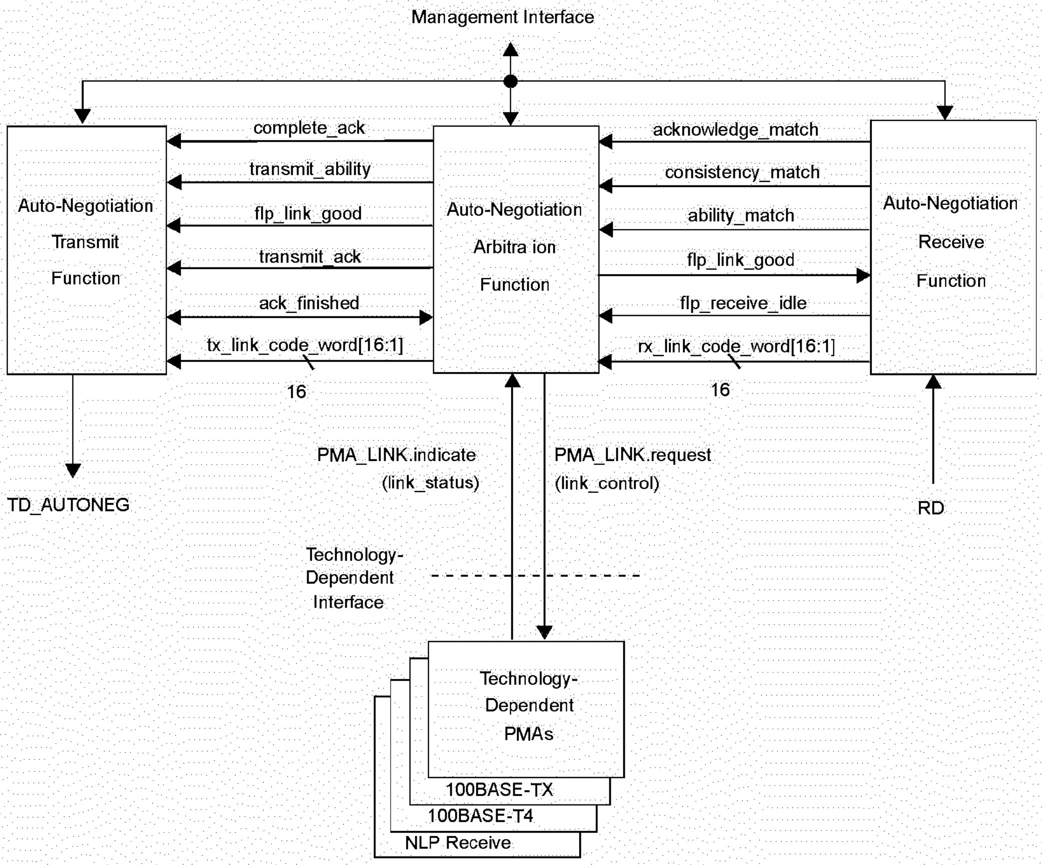


**Figure 28-13—Functional reference diagram**

### 28.3.1 State diagram variables

A variable with "_[x]" appended to the end of the variable name indicates a variable or set of variables as defined by "x". "x" may be as follows:

all; represents all specific technology-dependent PMAs supported in the Local Device and the NLP

Receive Link Integrity Test state diagram.

HCD; represents the single technology-dependent PMA chosen by Auto-Negotiation as the highest common denominator technology through the Priority Resolution or Parallel Detection function. To select 10BASE-T, LIT is used instead of NLP to enable the full 10BASE-T Link Integrity Test function state diagram.

notHCD; represents all technology-dependent PMAs not chosen by Auto-Negotiation as the highest common denominator technology through the Priority Resolution or Parallel Detection function.

TX; represents that the 100BASE-TX PMA is the signal source.

T4; represents that the 100BASE-T4 PMA is the signal source.

NLP; represents that the NLP Receive Link Integrity Test function is the signal source.

PD; represents all of the following that are present: 100BASE-TX PMA, 100BASE-T4 PMA, and the NLP Receive Link Integrity Test state diagram.

LIT; represents the 10BASE-T Link Integrity Test function state diagram is the signal source or destination.

Variables with [16:1] appended to the end of the variable name indicate arrays that can be directly mapped to 16-bit registers. For these variables, "[x]" indexes an element or set of elements in the array, where "[x]" may be as follows:

- Any integer.
- Any variable that takes on integer values.
- NP; represents the index of the Next Page bit.
- ACK; represents the index of the Acknowledge bit.
- RF; represents the index of the Remote Fault bit.

Variables of the form "mr_x", where x is a label, comprise a management interface that is intended to be connected to the MII Management function. However, an implementation-specific management interface may provide the control and status function of these bits.

ability_match

Indicates that three consecutive Link Code Words match, ignoring the Acknowledge bit. Three consecutive words are any three words received one after the other, regardless of whether the word has already been used in a word-match comparison or not.

Values: false; three matching consecutive Link Code Words have not been received, ignoring the Acknowledge bit (default).
true; three matching consecutive Link Code Words have been received, ignoring the Acknowledge bit.

NOTE—This variable is set by this variable definition; it is not set explicitly in the state diagrams.

ack_finished

Status indicating that the final remaining_ack_cnt Link Code Words with the Ack bit set have been transmitted.

Values: false; more Link Code Words with the Ack bit set to logic one must be transmitted.
true; all remaining Link Code Words with the Ack bit set to logic one have been transmitted.

acknowledge_match

Indicates that three consecutive Link Code Words match and have the Acknowledge bit set. Three consecutive words are any three words received one after the other, regardless of whether the word has already been used in a word match comparison or not.

Values: false; three matching and consecutive Link Code Words have not been received with the

Acknowledge bit set (default).
true; three matching and consecutive Link Code Words have been received with the Acknowledge bit set.

NOTE—This variable is set by this variable definition; it is not set explicitly in the state diagrams.

base_page

Status indicating that the page currently being transmitted by Auto-Negotiation is the initial Link Code Word encoding used to communicate the device's abilities.

Values: false; a page other than base Link Code Word is being transmitted.
true; the base Link Code Word is being transmitted.

complete_ack

Controls the counting of transmitted Link Code Words that have their Acknowledge bit set.

Values: false; transmitted Link Code Words with the Acknowledge bit set are not counted (default).
true; transmitted Link Code Words with the Acknowledge bit set are counted.

consistency_match

Indicates that the Link Code Word that caused ability_match to be set is the same as the Link Code Word that caused acknowledge_match to be set.

Values: false; the Link Code Word that caused ability_match to be set is not the same as the Link Code Word that caused acknowledge_match to be set, ignoring the Acknowledge bit value.
true; the Link Code Word that caused ability_match to be set is the same as the Link Code Word that caused acknowledge_match to be set, independent of the Acknowledge bit value.

NOTE—This variable is set by this variable definition; it is not set explicitly in the state diagrams.

desire_np

Status indicating that the Local Device desires to engage in Next Page exchange. This information comes from the setting of the NP bit in the base Link Code Word stored in the Auto-Negotiation advertisement register (register 4).

Values: false; Next Page exchange is not desired.
true; Next Page exchange is desired.

flp_link_good

Indicates that Auto-Negotiation has completed.

Values: false; negotiation is in progress (default).
true; negotiation is complete, forcing the Transmit and Receive functions to IDLE.

flp_receive_idle

Indicates that the Receive state diagram is in the IDLE, LINK PULSE DETECT, or LINK PULSE COUNT state.

Values: false; the Receive state diagram is not in the IDLE, LINK PULSE DETECT, or LINK PULSE COUNT state (default).
true; the Receive state diagram is in the IDLE, LINK PULSE DETECT, or LINK PULSE COUNT state.

link_control

This variable is defined in 28.2.6.2.1.

link_status

This variable is defined in 28.2.6.1.1.

linkpulse

Indicates that a valid Link Pulse as transmitted in compliance with figure 14-12 has been received.

Values: false; linkpulse is set to false after any Receive State Diagram state transition (default).

true; linkpulse is set to true when a valid Link Pulse is received.

mr_autoneg_complete

Status indicating whether Auto-Negotiation has completed or not.

Values: false; Auto-Negotiation has not completed.
true; Auto-Negotiation has completed.

mr_autoneg_enable

Controls the enabling and disabling of the Auto-Negotiation function.

Values: false; Auto-Negotiation is disabled.
true; Auto-Negotiation is enabled.

mr_adv_ability[16:1]

A 16-bit array that contains the Advertised Abilities Link Code Word.
For each element within the array:

Values: Zero; data bit is logical zero.
One; data bit is logical one.

mr_lp_adv_ability[16:1]

A 16-bit array that contains the Link Partner's Advertised Abilities Link Code Word.
For each element within the array:

Values: Zero; data bit is logical zero.
One; data bit is logical one.

mr_lp_np_able

Status indicating whether the Link Partner supports Next Page exchange.

Values: false; the Link Partner does not support Next Page exchange.
true; the Link Partner supports Next Page exchange.

mr_np_able

Status indicating whether the Local Device supports Next Page exchange.

Values: false; the Local Device does not support Next Page exchange.
true; the Local Device supports Next Page exchange.

mr_lp_autoneg_able

Status indicating whether the Link Partner supports Auto-Negotiation.

Values: false; the Link Partner does not support Auto-Negotiation.
true; the Link Partner supports Auto-Negotiation.

mr_main_reset

Controls the resetting of the Auto-Negotiation state diagrams.

Values: false; do not reset the Auto-Negotiation state diagrams.
true; reset the Auto-Negotiation state diagrams.

mr_next_page_loaded

Status indicating whether a new page has been loaded into the Auto-Negotiation Next Page Transmit register (register 7).

Values: false; a New Page has not been loaded.
true; a New Page has been loaded.

mr_np_tx[16:1]

A 16-bit array that contains the new Next Page to transmit.
For each element within the array:

Values: Zero; data bit is logical zero.
One; data bit is logical one.

mr_page_rx

Status indicating whether a New Page has been received. A New Page has been successfully received when acknowledge_match=true and consistency_match=true and the Link Code Word has been written to mr_lp_adv_ability[16:1].

Values: false; a New Page has not been received.
true; a New Page has been received.

mr_parallel_detection_fault

Error condition indicating that while performing Parallel Detection, either flp_receive_idle = false, or zero or more than one of the following indications were present when the autoneg_wait_timer expired. This signal is cleared on read of the Auto-Negotiaion expansion register.

1) link_status_ [NLP] = READY
2) link_status_[TX] = READY
3) link_status_[T4] = READY

Values: false; Exactly one of the above three indications was true when the autoneg_wait_timer expired, and flp_receive_idle = true.
true; either zero or more than one of the above three indications was true when the autoneg_wait_timer expired, or flp_receive_idle = false.

mr_restart_negotiation

Controls the entrance to the TRANSMIT DISABLE state to break the link before Auto-Negotiation is allowed to renegotiate via management control.

Values: false; renegotiation is not taking place.
true; renegotiation is started.

power_on

Condition that is true until such time as the power supply for the device that contains the Auto-Negotiation state diagrams has reached the operating region or the device has low power mode set via MII control register bit 0.11.

Values: false; the device is completely powered (default).
true; the device has not been completely powered.

rx_link_code_word[16:1]

A 16-bit array that contains the data bits to be received from an FLP Burst.
For each element within the array:

Values: zero; data bit is a logical zero.
one; data bit is a logical one.

single_link_ready

Status indicating that flp_receive_idle = true and only one the of the following indications is being received:

1) link_status_[NLP] = READY
2) link_status_[TX] = READY
3) link_status_[T4] = READY

Values: false; either zero or more than one of the above three indications are true or flp_receive_idle = false.
true; Exactly one of the above three indications is true and flp_receive_idle = true.

NOTE—This variable is set by this variable definition; it is not set explicitly in the state diagrams.

TD_AUTONEG

Controls the signal sent by Auto-Negotiation on the TD_AUTONEG circuit.

Values: idle; Auto-Negotiation prevents transmission of all link pulses on the MDI.
link_test_pulse; Auto-Negotiation causes a single link pulse as defined by figure 14-12 to be transmitted on the MDI.

toggle_rx

Flag to keep track of the state of the Link Partner's Toggle bit.

Values: 0; Link Partner's Toggle bit equals logic zero.
1; Link Partner's Toggle bit equals logic one.

toggle_tx

Flag to keep track of the state of the Local Device's Toggle bit.

Values: 0; Local Device's Toggle bit equals logic zero.
1; Local Device's Toggle bit equals logic one.

transmit_ability

Controls the transmission of the Link Code Word containing tx_link_code_word[16:1].

Values: false; any transmission of tx_link_code_word[16:1] is halted (default).
true; the transmit state diagram begins sending tx_link_code_word[16:1].

transmit_ack

Controls the setting of the Acknowledge bit in the tx_link_code_word[16:1] to be transmitted.

Values: false; sets the Acknowledge bit in the transmitted tx_link_code_word[16:1] to a logic zero (default).
true; sets the Acknowledge bit in the transmitted tx_link_code_word[16:1] to a logic one.

transmit_disable

Controls the transmission of tx_link_code_word[16:1].

Values: false; tx_link_code_word[16:1] transmission is allowed (default).
true; tx_link_code_word[16:1] transmission is halted.

tx_link_code_word[16:1]

A 16-bit array that contains the data bits to be transmitted in an FLP Burst. This array may be loaded from mr_adv_ability or mr_np_tx.
For each element within the array:

Values: Zero; data bit is logical zero.
One; data bit is logical one.

### 28.3.2 State diagram timers

All timers operate in the manner described in 14.2.3.2.

autoneg_wait_timer

Timer for the amount of time to wait before evaluating the number of link integrity test functions with link_status=READY asserted. The autoneg_wait_timer shall expire 500–1000 ms from the assertion of link_status=READY from the 100BASE-TX PMA, 100BASE-T4 PMA, or the NLP Receive State diagram.

break_link_timer

Timer for the amount of time to wait in order to assure that the Link Partner enters a Link Fail state. The timer shall expire 1200–1500 ms after being started.

data_detect_max_timer

Timer for the maximum time between a clock pulse and the next link pulse. This timer is used in conjunction with the data_detect_min_timer to detect whether the data bit between two clock pulses is a logic zero or a logic one. The data_detect_max_timer shall expire 78–100 μs from the last clock pulse.

data_detect_min_timer

Timer for the minimum time between a clock pulse and the next link pulse. This timer is used in conjunction with the data_detect_max_timer to detect whether the data bit between two clock pulses is a logic zero or a logic one. The data_detect_min_timer shall expire 15–47 μs from the last clock pulse.

flp_test_max_timer

Timer for the maximum time between two link pulses within an FLP Burst. This timer is used in conjunction with the flp_test_min_timer to detect whether the Link Partner is transmitting FLP Bursts. The flp_test_max_timer shall expire 165–185 μs from the last link pulse.

flp_test_min_timer

Timer for the minimum time between two link pulses within an FLP Burst. This timer is used in conjunction with the flp_test_max_timer to detect whether the Link Partner is transmitting FLP Bursts. The flp_test_min_timer shall expire 5–25 μs from the last link pulse.

interval_timer

Timer for the separation of a transmitted clock pulse from a data bit. The interval_timer shall expire 55.5–-69.5 μs from each clock pulse and data bit.

link_fail_inhibit_timer

Timer for qualifying a link_status=FAIL indication or a link_status=READY indication when a specific technology link is first being established. A link will only be considered "failed" if the link_fail_inhibit_timer has expired and the link has still not gone into the link_status=OK state. The link_fail_inhibit_timer shall expire 750–1000 ms after entering the FLP LINK GOOD CHECK state.

NOTE—The link_fail_inhibit_timer expiration value must be greater than the time required for the Link Partner to complete Auto-Negotiation after the Local Device has completed Auto-Negotiation plus the time required for the specific technology to enter the link_status=OK state. The maximum time difference between a Local Device and its Link Partner completing Auto-Negotiation is

(Maximum FLP Burst to FLP Burst separation) × (Maximum number of FLP Bursts needed to complete acknowledgment) = (24 ms) × (8 bursts) = 192 ms.

For example, 100BASE-T4 requires approximately 460 ms to enter link_status=OK for a total minimum link_fail_inhibit_timer time of 652 ms. The lower bound for the link_fail_inhibit_timer was chosen to provide adequate margin for the current technologies and any future PMAs.

nlp_test_max_timer

Timer for the maximum time that no FLP Burst may be seen before forcing the receive state diagram to the IDLE state. The nlp_test_max_timer shall expire 50–150 ms after being started or restarted.

nlp_test_min_timer

Timer for the minimum time between two consecutive FLP Bursts. The nlp_test_min_timer shall expire 5–7 ms after being started or restarted.

transmit_link_burst_timer

Timer for the separation of a transmitted FLP Burst from the next FLP Burst. The transmit_link_burst_timer shall expire 5.7-22.3 ms after the last transmitted link pulse in an FLP Burst.

### Table 28-8—Timer min./max. value summary

| Parameter | Min. | Typ. | Max. | Units |
|---|---|---|---|---|
| autoneg_wait_timer | 500 | | 1000 | ms |
| break_link_timer | 1200 | | 1500 | ms |
| data_detect_min_timer | 15 | | 47 | μs |
| data_detect_max_timer | 78 | | 100 | μs |

**Table 28-8—Timer min./max. value summary *(Continued)***

| Parameter | Min. | Typ. | Max. | Units |
|---|---|---|---|---|
| flp_test_min_timer | 5 | | 25 | μs |
| flp_test_max_timer | 165 | | 185 | μs |
| interval_timer | 55.5 | 62.5 | 69.5 | μs |
| link_fail_inhibit_timer | 750 | | 1000 | ms |
| nlp_test_max_timer | 50 | | 150 | ms |
| nlp_test_min_timer | 5 | | 7 | ms |
| transmit_link_burst_timer | 5.7 | 14 | 22.3 | ms |

### 28.3.3 State diagram counters

flp_cnt

A counter that may take on integer values from 0 to 17. This counter is used to keep a count of the number of FLPs detected to enable the determination of whether the Link Partner supports Auto-Negotiation.

Values: not_done; 0 to 5 inclusive.
done; 6 to 17 inclusive.
init; counter is reset to zero.

remaining_ack_cnt

A counter that may take on integer values from 0 to 8. The number of additional Link Code Words with the Acknowledge Bit set to logic one to be sent to ensure that the Link Partner receives the acknowledgment.

Values: not_done; positive integers between 0 and 5 inclusive.
done; positive integers 6 to 8 inclusive (default).
init; counter is reset to zero.

rx_bit_cnt

A counter that may take on integer values from 0 to 17. This counter is used to keep a count of data bits received from an FLP Burst and to ensure that when erroneous extra pulses are received, the first 16 bits are kept while the rest are ignored. When this variable reaches 16 or 17, enough data bits have been received. This counter does not increment beyond 17 and does not return to 0 until it is reinitialized.

Values: not_done; 1 to 15 inclusive.
done; 16 or 17
init; counter is reset to zero.
rx_bit_cnt_check; 10 to 17 inclusive.

tx_bit_cnt

A counter that may take on integer values from 1 to 17. This counter is used to keep a count of data bits sent within an FLP Burst. When this variable reaches 17, all data bits have been sent.

Values: not_done; 1 to 16 inclusive.
done; 17.
init; counter is initialized to 1.

### 28.3.4 State diagrams

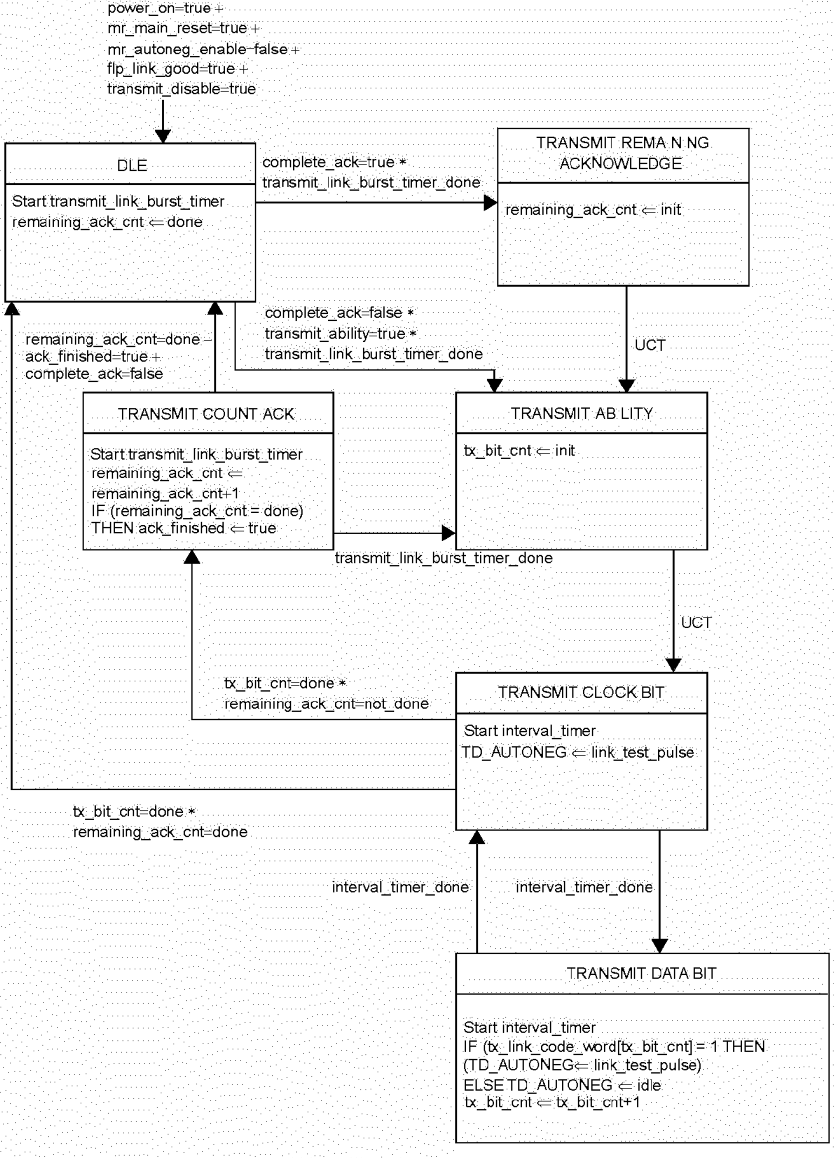


Figure 28-14—Transmit state diagram

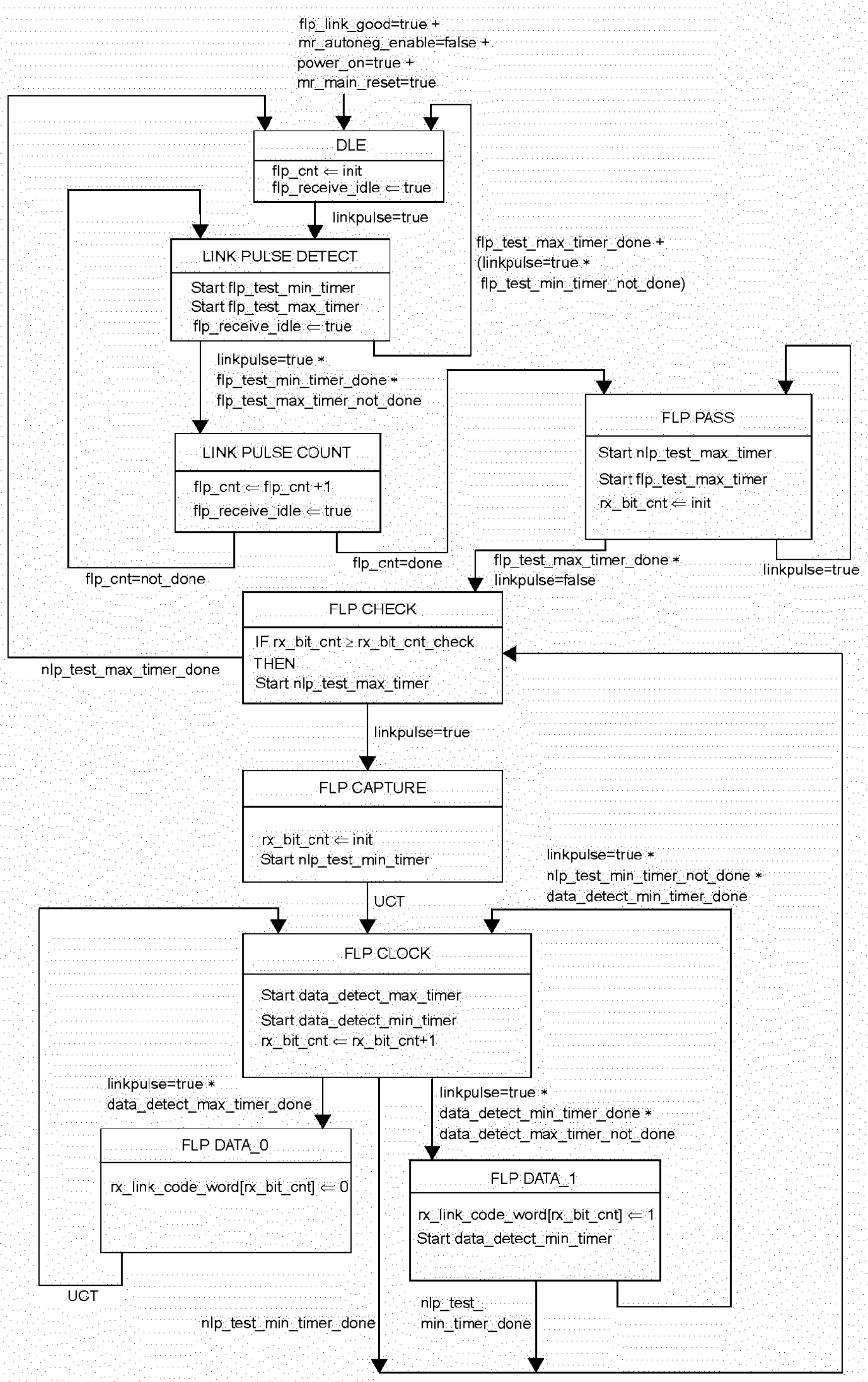
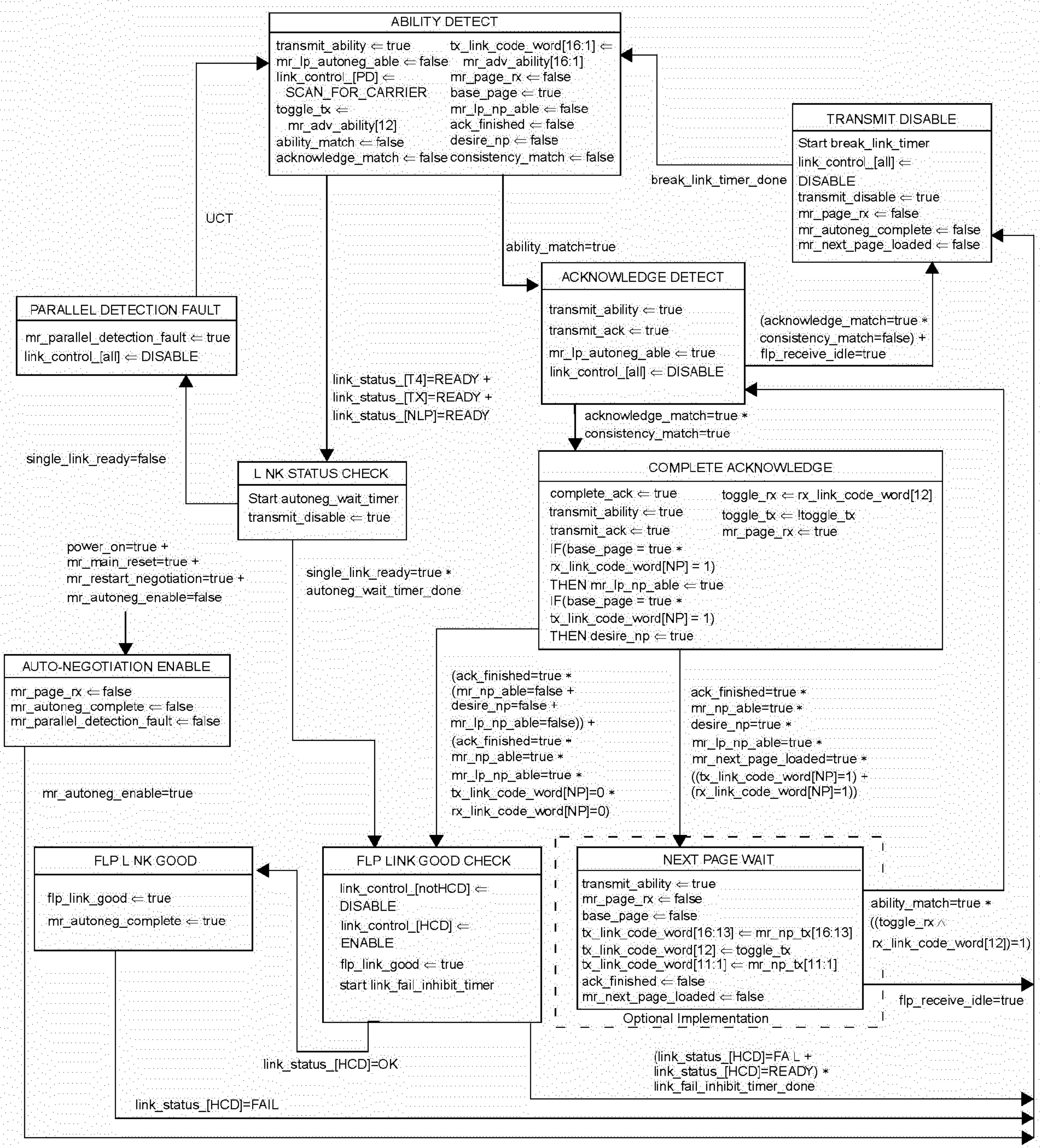


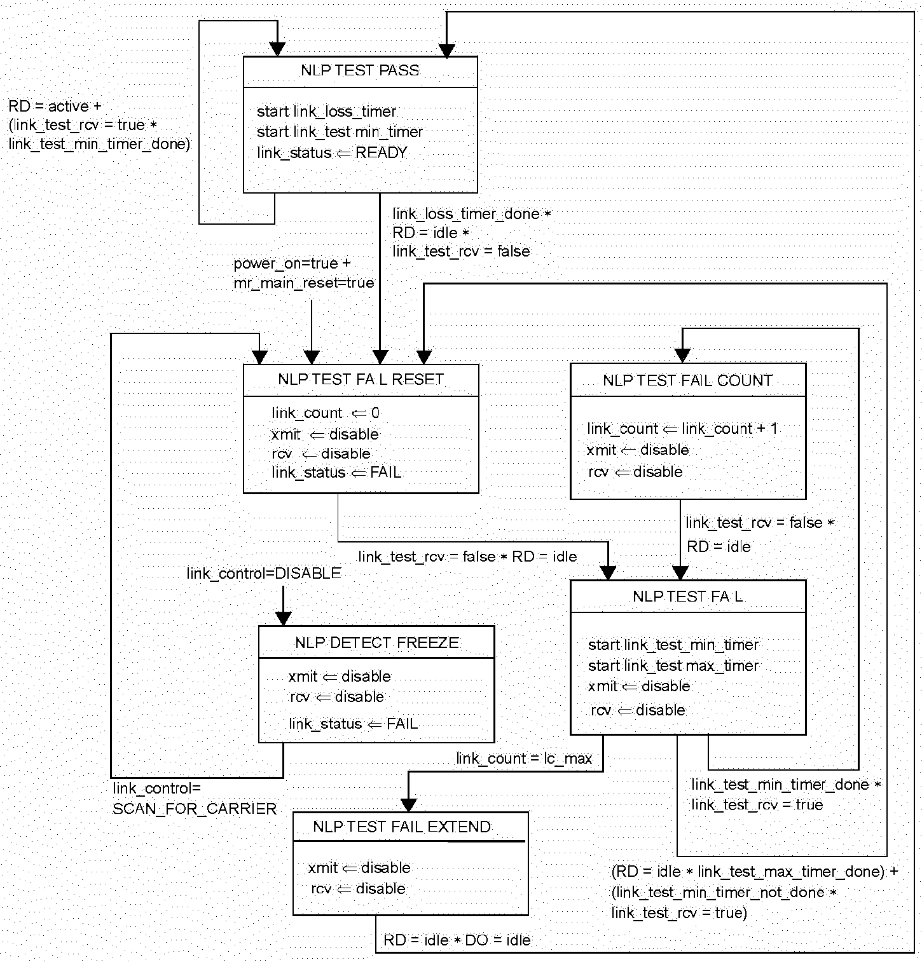
Figure 28-15—Receive state diagram



NOTE—The transition from COMPLETE ACKNOWLEDGE to FLP LINK GOOD CHECK can be simplified to "ack_finished=true" if the optional Next Page function is not supported.

NOTE—ability_match, acknowledge_match, single_link_ready, and consistency_match are set according to the variable definitions and are not set explicitly in the state diagrams.

**Figure 28-16—Arbitration state diagram**



NOTE—The variables link_control and link_status are viewed as dedicated signals by the NLP Receive Link integrity Test state diagram, but are viewed as link_control_[NLP] and link_status_[NLP] by the Auto-Negotiation Arbitration state diagram, figure 28-16.

**Figure 28-17—NLP Receive Link Integrity Test state diagram**

## 28.4 Electrical specifications

The electrical characteristics of pulses within FLP Bursts shall be identical to the characteristics of NLPs and shall meet the requirements of figure 14-12.

It is the responsibility of the technology-specific Transmit and Receive functions to interface to the MDI correctly.

NOTE—The requirements relative to the interface to the MDI are specified via the Transmit Switch and Receive Switch functions.

## 28.5 Protocol Implementation Conformance Statement (PICS) proforma for clause 28, Physical Layer link signaling for 10 Mb/s and 100 Mb/s Auto-Negotiation on twisted pair[27]

### 28.5.1 Introduction

The supplier of a protocol implementation that is claimed to conform to IEEE Std 802.3u-1995, Physical Layer link signaling for 10 Mb/s and 100 Mb/s Auto-Negotiation on twisted pair, shall complete the following Protocol Implementation Conformance Statement (PICS) proforma.

A detailed description of the symbols used in the PICS proforma, along with instructions for completing the PICS proforma, can be found in clause 21.

### 28.5.2 Identification

#### 28.5.2.1 Implementation identification

| | |
|---|---|
| Supplier | |
| Contact point for enquiries about the PICS | |
| Implementation Name(s) and Version(s) | |
| Other information necessary for full identification—e.g., name(s) and version(s) for machines and/or operating systems; System Names(s) | |
| NOTES<br>1—Only the first three items are required for all implementations; other information may be completed as appropriate in meeting the requirements for the identification.<br>2—The terms Name and Version should be interpreted appropriately to correspond with a supplier's terminology (e.g., Type, Series, Model). | |

#### 28.5.2.2 Protocol summary

| | |
|---|---|
| Identification of protocol standard | IEEE Std 802.3u-1995, Physical Layer link signaling for 10 Mb/s and 100 Mb/s Auto-Negotiation on twisted pair |
| Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS | |
| Have any Exception items been required? No [ ] Yes [ ]<br>(See clause 21; the answer Yes means that the implementation does not conform to IEEE Std 802.3u-1995.) | |
| Date of Statement | |

---

[27] *Copyright release for PICS proformas* Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

### 28.5.3 Major capabilities/options

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 10BT | Implementation supports a 10BASE-T data service | 28.1.2 | O | | N/A |
| *NP | Implementation supports Next Page function | 28.1.2 | O | | N/A |
| *MII | Implementation supports the MII Management Interface | 28.1.2 | O/1 | | N/A |
| MGMT | Implementation supports a non-MII Management Interface | 28.1.2 | O/1 | | N/A |
| *NOM | Implementation does not support management | 28.1.2 | O/1 | | N/A |
| *RF | Implementation supports Remote Fault Sensing | 28.2.3.5 | O | | N/A |

### 28.5.4 PICS proforma tables for Physical Layer link signaling for 10 Mb/s and 100 Mb/s Auto-Negotiation on twisted pair

#### 28.5.4.1 Scope

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 1 | MII Management Interface control and status registers | 28.1.3 | MII:M | | Implemented in accordance with the definitions in clause 22 and 28.2.4 |
| 2 | CSMA/CD compatible devices using an eight-pin modular connector and using a signaling method to automatically configure the preferred mode of operation | 28.1.4 | M | | Auto-Negotiation function implemented in compliance with clause 28 |
| 3 | Device uses 10BASE-T compatible link signaling to advertise non-CSMA/CD abilities | 28.1.4 | M | | Auto-Negotiation function implemented in compliance with clause 28 |
| 4 | Future CSMA/CD implementations that use an eight-pin modular connector | 28.1.4 | M | | Interoperable with devices compliant with clause 28 |

### 28.5.4.2 Auto-Negotiation functions

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 1 | Transmit | 28.2 | M | | Complies with figure 28-14 |
| 2 | Receive | 28.2 | M | | Complies with figure 28-15 |
| 3 | Arbitration | 28.2 | M | | Complies with figure 28-16 |
| 4 | NLP Receive Link Integrity Test | 28.2 | M | | Complies with figure 28-17 |
| 5 | Technology-Dependent Interface | 28.2 | M | | Complies with 28.2.6 |
| 6 | Technology-dependent link integrity test | 28.2 | M | | Implemented and interfaced to for those technologies supported by device |
| 7 | Management | 28.2 | O | | MII based or alternate management |

### 28.5.4.3 Transmit function requirements

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 1 | FLP Burst transmission | 28.2.1.1 | M | | Not transmitted once Auto-Negotiation is complete and highest common denominator PMA has been enabled. Prohibited other than for link start-up |
| 2 | FLP Burst composition | 28.2.1.1.1 | M | | Pulses in FLP Bursts meet the requirements of figure 14-12 |
| 3 | FLP Burst pulse definition | 28.2.1.1.1 | M | | 17 odd-numbered pulse positions represent clock information; 16 even-numbered pulse positions represent data information |
| 4 | The first pulse in an FLP Burst | 28.2.1.1.2 | M | | Defined as a clock pulse for timing purposes |
| 5 | FLP Burst clock pulse spacing | 28.2.1.1.2 | M | | Within an FLP Burst, spacing is 125 ± 14 µs |
| 6 | Logic one data bit representation | 28.2.1.1.2 | M | | Pulse transmitted 62.5 ± 7 ∝s after the preceding clock pulse |
| 7 | Logic zero data bit representation | 28.2.1.1.2 | M | | No link integrity test pulses within 111 µs of the preceding clock pulse |
| 8 | Consecutive FLP Bursts | 28.2.1.1.2 | M | | The first link pulse in each FLP Burst is separated by 16 ± 8 ms |
| 9 | FLP Burst base page | 28.2.1.2 | M | | Conforms to figure 28-7 |

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 10 | FLP Burst bit transmission order | 28.2.1.2 | M | | Transmission is D0 first to D15 last |
| 11 | Selector Field values | 28.2.1.2.1 | M | | Only defined values transmitted |
| 12 | Technology Ability Field values | 28.2.1.2.2 | M | | Implementation supports a data service for each ability set in the Technology Ability Field |
| 13 | Remote Fault bit | 28.2.1.2.3 | M | | Used in accordance with the Remote Fault function specifications |
| 14 | Acknowledge bit set, no Next Page to be sent | 28.2.1.2.4 | M | | Set to logic one in the Link Code Word after the reception of at least three consecutive and consistent FLP Bursts |
| 15 | Acknowledge bit set, Next Page to be sent | 28.2.1.2.4 | NP:M | | Set to logic one in the transmitted Link Code Word after the reception of at least three consecutive and consistent FLP Bursts and the current receive Link Code Word is saved |
| 16 | Number of Link Code Words sent with Acknowledge bit set | 28.2.1.2.4 | M | | 6 to 8 inclusive after COMPLETE ACKNOWLEDGE state entered |
| 17 | Device does not implement optional Next Page ability | 28.2.1.2.5 | M | | NP=0 in base Link Code Word |
| 18 | Device implements optional Next Page ability and wishes to engage in Next Page exchange | 28.2.1.2.5 | NP:M | | NP=1 in base Link Code Word |
| 19 | Transmit Switch function on completion of Auto-Negotiation | 28.2.1.3 | M | | Enables the transmit path from a single technology-dependent PMA to the MDI once the highest common denominator has been selected |
| 20 | Transmit Switch function during Auto-Negotiation | 28.2.1.3 | M | | Connects FLP Burst generator governed by figure 28-14 to the MDI |
| 21 | Signals presented at MDI after connection through Transmit Switch from PMA | 28.2.1.3 | M | | Conform to appropriate PHY specifications |

### 28.5.4.4 Receive function requirements

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 1 | Timer expiration | 28.2.2.1 | M | | Timer definition in 28.3.2, values shown in table 28-8 |
| 2 | Identification of Link Partner as Auto-Negotiation able | 28.2.2.1 | M | | Reception of 6 to 17 (inclusive) consecutive link pulses separated by at least flp_test_min_timer time but less than flp_test_max_timer time |
| 3 | First FLP Burst identifying Link Partner as Auto-Negotiation able | 28.2.2.1 | M | | Data recovered is discarded if FLP Burst is incomplete |
| 4 | First link pulse in an FLP Burst | 28.2.2.1 | M | | Interpreted as a clock link pulse |
| 5 | Restart of the data_detect_min_timer and data_detect_max_timer | 28.2.2.1 | M | | Detection of a clock link pulse (figure 28-9) |
| 6 | Reception of logic one | 28.2.2.1 | M | | Link pulse received between greater than data_detect_min_timer time and less than data_detect_max_timer time after a clock pulse (figure 28-9) |
| 7 | Reception of logic zero | 28.2.2.1 | M | | Link pulse received after greater than data_detect_max_timer time after clock pulse, is treated as clock pulse (figure 28-9) |
| 8 | FLP Bursts separation | 28.2.2.1 | M | | Conforms to the nlp_test_min_timer and nlp_test_max_timer timing (figure 28-10) |
| 9 | Receive Switch function on completion of Auto-Negotiation | 28.2.2.3 | M | | Enables the receive path from the MDI to a single technology-dependent PMA once the highest common denominator has been selected |
| 10 | Receive Switch function during Auto-Negotiation | 28.2.2.3 | M | | Connects the MDI to the FLP and NLP receivers governed by figures 28-15 and 28-17, and to the 100BASE-TX and 100BASE-T4 receivers if present |
| 11 | Signals presented to PMA after connection through Receive Switch from MDI | 28.2.2.3 | M | | Conform to appropriate PHY specifications |
| 12 | Generation of ability_match, acknowledge_match, and consistency_match | 28.2.2.4 | M | | Responsibility of Receive function in accordance with 28.3.1 |

#### 28.5.4.5 Arbitration functions

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 1 | MDI receive connection during Auto-Negotiation, prior to FLP detection | 28.2.3.1 | M | | Connected to the NLP Receive Link Integrity Test state diagram, and the link integrity test functions of 100BASE-TX and/or 100BASE-T4. Not connected to the 10BASE-T or any other PMA |
| 2 | Parallel detection operational mode selection | 28.2.3.1 | M | | Set link_control=ENABLE for the single PMA indicating link_status=READY when the autoneg_wait_timer expires |
| 3 | Parallel detection PMA control | 28.2.3.1 | M | | Set link_control=DISABLE to all PMAs except the selected operational PMA and indicate Auto-Negotiation has completed |
| 4 | Parallel detection setting of link partner ability register | 28.2.3.1 | M | | On transition to the FLP LINK GOOD CHECK state from the LINK STATUS CHECK state the Parallel Detection function shall set the bit in the link partner ability register (register 5) corresponding to the technology detected by the Parallel Detection function |
| 5 | Response to renegotiation request | 28.2.3.2 | M | | Disable all technology-dependent link integrity test functions and halt transmit activity until break_link_timer expires |
| 6 | Auto-Negotiation resumption | 28.2.3.2 | M | | Issue FLP Bursts with base page valid in tx_link_code_word[16:1] after break_link_timer expires |
| 7 | Priority resolution | 28.2.3.3 | M | | Single PMA connected to MDI is enabled corresponding to Technology Ability Field bit common to both Local/Link Partner Device and that has highest priority as defined by annex 28B |
| 8 | Effect of receipt of reserved Technology Ability Field bit on priority resolution | 28.2.3.3 | M | | Local Device ignores during priority resolution |
| 9 | Effect of parallel detection on priority resolution | 28.2.3.3 | M | | Local Device considers technology identified by parallel detection as HCD |
| 10 | Values for HCD and link_status_[HCD] in the event there is no common technology | 28.2.3.3 | M | | HCD=NULL<br>link_status_[HCD]=FAIL |

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 11 | Message Page to Unformatted Page relationship for non-matching Selector Fields | 28.2.3.4 | NP:M | | Each series of Unformatted Pages is preceded by an Message Page containing a message code that defines how the following Unformatted Page(s) will be interpreted |
| 12 | Message Page to Unformatted Page relationship for matching Selector Fields | 28.2.3.4 | NP:M | | Use of Message Pages is specified by the Selector Field value |
| 13 | Transmission of Null message codes | 28.2.3.4 | NP:M | | Sent with NP=0 on completion of all Next Pages while Link Partner continues to transmit valid Next Page information |
| 14 | Reception of Null message codes | 28.2.3.4 | NP:M | | Recognized as indicating end of Link Partner's Next Page information |
| 15 | Next Page encoding | 28.2.3.4.1 | NP:M | | Comply with figures 28-11 and 28-12 for the NP, Ack, MP, Ack2, and T bits |
| 16 | Message/Unformatted Code Field | 28.2.3.4.1 | NP:M | | D10-D0 encoded as Message Code Field if MP=1 or Unformatted Code Field if MP=0 |
| 17 | NP bit encoding | 28.2.3.4.2 | NP:M | | Logic 0=last page, logic 1=additional Next Page(s) follow |
| 18 | Message Page bit encoding | 28.2.3.4.4 | NP:M | | Logic 0=Unformatted Page, logic 1=Message Page |
| 19 | Ack2 bit encoding | 28.2.3.4.5 | NP:M | | Logic 0=cannot comply with message; logic 1= will comply with message |
| 20 | Toggle | 28.2.3.4.6 | NP:M | | Takes the opposite value of the Toggle bit in the previously exchanged Link Code Word |
| 21 | Toggle encoding | 28.2.3.4.6 | NP:M | | Logic zero = previous value of the transmitted Link Code Word equalled logic one<br>Logic one = previous value of the transmitted Link Code Word equalled logic zero |
| 22 | Message Page encoding | 28.2.3.4.7 | NP:M | | If MP=1, Link Code Word interpreted as Message Page |
| 23 | Message Code Field | 28.2.3.4.8 | NP:M | | Combinations not shown in annex 28B are reserved and may not be transmitted |
| 24 | Unformatted Page encoding | 28.2.3.4.9 | NP:M | | If MP=0, Link Code Word interpreted as Unformatted Page |

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 25 | Minimum Next Page exchange | 28.2.3.4.11 | NP:M | | If both devices indicate Next Page able, both send a minimum of one Next Page |
| 26 | Multiple Next Page exchange | 28.2.3.4.11 | NP:M | | If both devices indicate Next Page able, exchange continues until neither Local/Remote Device has additional information; device sends Next Page with Null Message Code if it has no information to transmit |
| 27 | Unformatted Page ordering | 28.2.3.4.11 | NP:M | | Unformatted Pages immediately follow the referencing Message Code in the order specified by the Message Code |
| 28 | Next Page Transmit register | 28.2.3.4.12 | NP:M | | Defined in 28.2.4.1.6 |
| 29 | Next Page receive data | 28.2.3.4.12 | NP:O | | May be stored in Auto-Negotiation link partner ability register |
| 30 | Remote Fault sensing | 28.2.3.5 | RF:M | | Optional |
| 31 | Transmission of RF bit by Local Device | 28.2.3.5 | M | | If Local Device has no method to set RF bit, it must transmit RF bit with value of RF bit in Auto-Negotiation advertisement register (4.13) |
| 32 | RF bit reset | 28.2.3.5 | M | | Once set, the RF bit remains set until successful renegotiation with the base Link Code Word |
| 33 | Receipt of Remote Fault indication in Base Link Code Word | 28.2.3.5 | MII:M | | Device sets the Remote Fault bit in the MII status register (1.4) to logic one if MII is present |

#### 28.5.4.6 Management function requirements

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 1 | Mandatory MII registers for Auto-Negotiation | 28.2.4.1 | MII:M | | Registers 0, 1, 4, 5, 6 |
| 2 | Optional MII register for Auto-Negotiation | 28.2.4.1 | MII*<br>NP:M | | Register 7 |
| 3 | Auto-Negotiation enable | 28.2.4.1.1 | MII:M | | Set control register Auto-Negotiation Enable bit (0.12) |
| 4 | Manual Speed/Duplex settings | 28.2.4.1.1 | MII:M | | When bit 0.12 set, control register Speed Detection (0.13) and Duplex Mode (0.8) are ignored, and the Auto-Negotiation function determines link configuration |
| 5 | control register (register 0) Restart Auto-Negotiation (0.9) default | 28.2.4.1.1 | MII:M | | PHY returns value of one in 0.9 until Auto-Negotiation has been initiated |
| 6 | control register (register 0) Restart Auto-Negotiation (0.9) set | 28.2.4.1.1 | MII:M | | When 0.9 set, Auto-Negotiation will (re)initiate. On completion, 0.9 will be reset by the PHY device. Writing a zero to 0.9 at any time has no effect |
| 7 | control register (register 0) Restart Auto-Negotiation (0.9) reset | 28.2.4.1.1 | MII:M | | 0.9 is self-clearing; writing a zero to 0.9 at any time has no effect |
| 8 | status register (register 1) Auto-Negotiation Complete (1.5) reset | 28.2.4.1.2 | MII:M | | If bit 0.12 reset, or a PHY lacks the ability to perform Auto-Negotiation, (1.5) is reset |
| 9 | status register (register 1) Remote Fault (1.4) | 28.2.4.1.2 | MII:M | | Set by the PHY and remains set until either the status register is read or the PHY is reset |
| 10 | advertisement register power on default | 28.2.4.1.3 | MII:M | | Selector field as defined in annex 28A; Ack=0; Technology Ability Field based on MII status register (1.15:11) or logical equivalent |
| 11 | link partner ability register read/write | 28.2.4.1.4 | MII:M | | Read only; write has no effect |
| 12 | link partner ability register bit definitions | 28.2.4.1.4 | MII:M | | Direct representation of the received Link Code Word (figure 28-7) |
| 13 | status register (register 1) Auto-Negotiation Complete (1.5) set | 28.2.4.1.4 | MII:M | | Set to logic one upon successful completion of Auto-Negotiation |
| 14 | Auto-Negotiation expansion register (register 6) | 28.2.4.1.5 | MII:M | | Read only; write has no effect |
| 15 | Link Partner Auto-Negotiation Able bit (6.0) | 28.2.4.1.5 | MII:M | | Set to indicate that the Link Partner is able to participate in the Auto-Negotiation function |

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 16 | Page Received bit (6.1) set | 28.2.4.1.5 | MII:M | | Set to indicate that a new Link Code Word has been received and stored in the Auto-Negotiation link partner ability register |
| 17 | Page Received bit (6.1) reset | 28.2.4.1.5 | MII:M | | Reset on a read of the Auto-Negotiation expansion register (register 6) |
| 18 | The Next Page Able bit (6.2) set | 28.2.4.1.5 | NP*<br>MII:M | | Set to indicate that the Local Device supports the Next Page function |
| 19 | The Link Partner Next Page Able bit (6.3) set | 28.2.4.1.5 | MII:M | | Set to indicate that the Link Partner supports the Next Page function |
| 20 | Parallel Detection Fault bit (6.4) set | 28.2.4.1.5 | MII:M | | Set to indicate that zero or more than one of the NLP Receive Link Integrity Test function, 100BASE-TX, or 100BASE-T4 PMAs have indicated link_status=READY when the autoneg_wait_timer expires |
| 21 | Parallel Detection Fault bit (6.4) reset | 28.2.4.1.5 | MII:M | | Reset on a read of the Auto-Negotiation expansion register (register 6) |
| 22 | Next Page Transmit register default | 28.2.4.1.6 | NP*<br>MII:M | | On power-up, contains value of 2001 H |
| 23 | Write to Next Page Transmit register | 28.2.4.1.6 | NP*<br>MII:M | | mr_next_page_loaded set to true |
| 24 | Absence of management function | 28.2.5 | NOM:M | | Advertised abilities provided through a logical equivalent of mr_adv_ability[16:1] |
| 25 | Next Page support in absence of MII management | 28.2.5 | NOM:M | | Device must provide logical equivalent of mr_np_able, mr_lp_np_able, or mr_next_page_loaded variables in order to set NP bit in transmitted Link Code Word |

### 28.5.4.7 Technology-dependent interface

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 1 | PMA_LINK.indicate (link_status) values | 28.2.6.1.1 | M | | link_status set to READY, OK or FAIL |
| 2 | PMA_LINK.indicate(link_status) generation | 28.2.6.1.2 | M | | Technology-dependent PMA and NLP Receive Link Integrity Test state diagram (figure 28-17) responsibility |
| 3 | PMA_LINK.indicate(link_status), effect of receipt | 28.2.6.1.3 | M | | Governed by the state diagram of figure 28-16 |
| 4 | PMA_LINK.request(link_control) values | 28.2.6.1.3 | M | | link_control set to SCAN_FOR_CARRIER, DISABLE, or ENABLE |
| 5 | Effect of link_control=SCAN_FOR_CARRIER | 28.2.6.2.1 | M | | PMA to search for carrier and report link_status=READY when carrier is received, but no other actions are enabled |
| 6 | Effect of link_control=DISABLE | 28.2.6.2.1 | M | | Disables PMA processing |
| 7 | Effect of link_control=ENABLE | 28.2.6.2.1 | M | | Control passed to a single PMA for normal processing functions |
| 8 | PMA_LINK.request(link_control) generation | 28.2.6.2.2 | M | | Auto-Negotiation function responsibility in accordance with figures 28-15 and 28-16 |
| 9 | PMA_LINK.request(link_control) default upon power-on, reset, or release from power-down | 28.2.6.2.2 | M | | link_control = DISABLE state to all technology-dependent PMAs |
| 10 | PMA_LINK.request(link_control) effect of receipt | 28.2.6.2.3 | M | | Governed by figure 28-17 and the receiving technology-dependent link integrity test function |

### 28.5.4.8 State diagrams

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 1 | Adherence to state diagrams | 28.3 | M | | Implement all features of figures 28-14 to 28-17. Identified options to figures 28-14 to 28-17 are permitted |
| 3 | Ambiguous requirements | 28.3 | M | | State diagrams take precedence in defining functional operation |
| 4 | autoneg_wait_timer | 28.3.1 | M | | Expires between 500–1000 ms after being started |
| 5 | break_link_timer | 28.3.2 | M | | Expires between 1200–1500 ms after being started |
| 6 | data_detect_min_timer | 28.3.2 | M | | Expires between 15–47 ∝s from the last clock pulse |
| 7 | data_detect_max_timer | 28.3.2 | M | | Expire between 78–100 ∝s from the last clock pulse |
| 8 | flp_test_max_timer | 28.3.2 | M | | Expires between 165–185 ∝s from the last link pulse |
| 9 | flp_test_min_timer | 28.3.2 | M | | Expires between 5–25 ∝s from the last link pulse |
| 10 | interval_timer | 28.3.2 | M | | Expires 55.5–69.5 ∝s from each clock pulse and data bit |
| 11 | link_fail_inhibit_timer | 28.3.2 | M | | Expires 750–1000 ms after entering the FLP LINK GOOD CHECK state |
| 12 | nlp_test_max_timer | 28.3.2 | M | | Expires between 50–150 ms after being started if not restarted |
| 13 | nlp_test_min_timer | 28.3.2 | M | | Expires between 5–7 ms after being started if not restarted |
| 14 | transmit_link_burst_timer | 28.3.1 | M | | Expires 5.7–22.3 ms after the last transmitted link pulse in an FLP Burst |

### 28.5.4.9 Electrical characteristics

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 1 | Pulses within FLP Bursts | 28.4 | M | | Identical to the characteristics of NLPs and meet the requirements of figure 14-12 |

### 28.5.4.10 Auto-Negotiation annexes

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 1 | Selector Field, S[4:0] values in the Link Code Word | 28A | M | | Identifies base message type as defined by table 28-9 |
| 2 | Selector Field reserved combinations | 28A | M | | Transmission not permitted |
| 3 | Relative priorities of the technologies supported by the IEEE 802.3 Selector Field value | 28B.3 | M | | Defined in 28B.3 |
| 4 | Relative order of the technologies supported by IEEE 802.3 Selector Field | 28B.3 | M | | Remain unchanged |
| 5 | Addition of new technology | 28B.3 | M | | Inserted into its appropriate place in the priority resolution hierarchy, shifting technologies of lesser priority lower in priority |
| 6 | Addition of vendor-specific technology | 28B.3 | M | | Priority of IEEE 802.3 standard topologies maintained, vendor-specific technologies to be inserted into an appropriate location |
| 7 | Message Code Field | 28C | NP:M | | Defines how following Unformatted Pages (if applicable) are interpreted |
| 8 | Message Code Field reserved combinations | 28C | NP:M | | Transmission not permitted |
| 9 | Auto-Negotiation reserved code 1 | 28C.1 | NP:M | | Transmission of M10 to M0 equals 0, not permitted |
| 10 | Null Message Code | 28C.2 | NP:M | | Transmitted during Next Page exchange when the Local Device has no information to transmit and Link Partner has additional pages to transmit |
| 11 | Remote Fault Identifier Message Code | 28C.5 | NP:M | | Followed by single Unformatted Page to identify fault type with types defined in 28C.5 |

| Item | Feature | Subclause | Status | Support | Value/comment |
|---|---|---|---|---|---|
| 12 | Organizationally Unique Identifier Message Code | 28C.6 | NP:M | | Followed by 4 Unformatted Pages. First Unformatted Page contains most significant 11 bits of OUI (bits 23:13) with MSB in U10;<br>Second Unformatted Page contains next most significant 11 bits of OUI (bits 12:2), with MSB in U10;<br>Third Unformatted Page contains the least significant 2 bits of OUI (bits 1:0) with MSB in U10, bits U8:0 contains user-defined code specific to OUI;<br>Fourth Unformatted Page contains user-defined code specific to OUI |
| 13 | PHY Identifier Message Code | 28C.7 | NP:M | | Followed by 4 Unformatted Pages. First Unformatted Page contains most significant 11 bits of PHY ID (2.15:5) with MSB in U10;<br>Second Unformatted Page contains PHY ID bits 2.4:0 to 3.15:10, with MSB in U10;<br>Third Unformatted Page contains PHY ID bits 3.9:0, with MSB in U10, and U0 contains user-defined code specific to PHY ID;<br>Fourth Unformatted Page contains user-defined code specific to PHY ID |
| 14 | Auto-Negotiation reserved code 2 | 28C.8 | NP:M | | Transmission of M10 to M0 equals 1, not permitted |

## 28.6 Auto-Negotiation expansion

Auto-Negotiation is designed in a way that allows it to be easily expanded as new technologies are developed. When a new technology is developed, the following things must be done to allow Auto-Negotiation to support it:

a) The appropriate Selector Field value to contain the new technology must be selected and allocated.
b) A Technology bit must be allocated for the new technology within the chosen Selector Field value.
c) The new technology's relative priority within the technologies supported within a Selector Field value must be established.

Code space allocations are enumerated in annexes 28A, 28B, and 28C. Additions and insertions to the annexes are allowed. No changes to existing bits already defined are allowed.

# 29. System considerations for multi-segment 100BASE-T networks

## 29.1 Overview

This clause provides information on building 100BASE-T networks. The 100BASE-T technology is designed to be deployed in both homogenous 100 Mb/s networks and heterogeneous 10/100 Mb/s mixed CSMA/CD networks. Network topologies can be developed within a single 100BASE-T collision domain, but maximum flexibility is achieved by designing multiple collision domain networks that are joined by bridges and/or routers configured to provide a range of service levels to DTEs. For example, a combined 100BASE-T/10BASE-T system built with repeaters and bridges can deliver dedicated 100 Mb/s, shared 100 Mb/s, dedicated 10 Mb/s, and shared 10 Mb/s service to DTEs. The effective bandwidth of shared services is controlled by the number of DTEs that share the service.

Linking multiple 100BASE-T collision domains with bridges maximizes flexibility. Bridged topology designs can provide single bandwidth (fignre 29-1) or multiple bandwidth (figure 29-2) services.

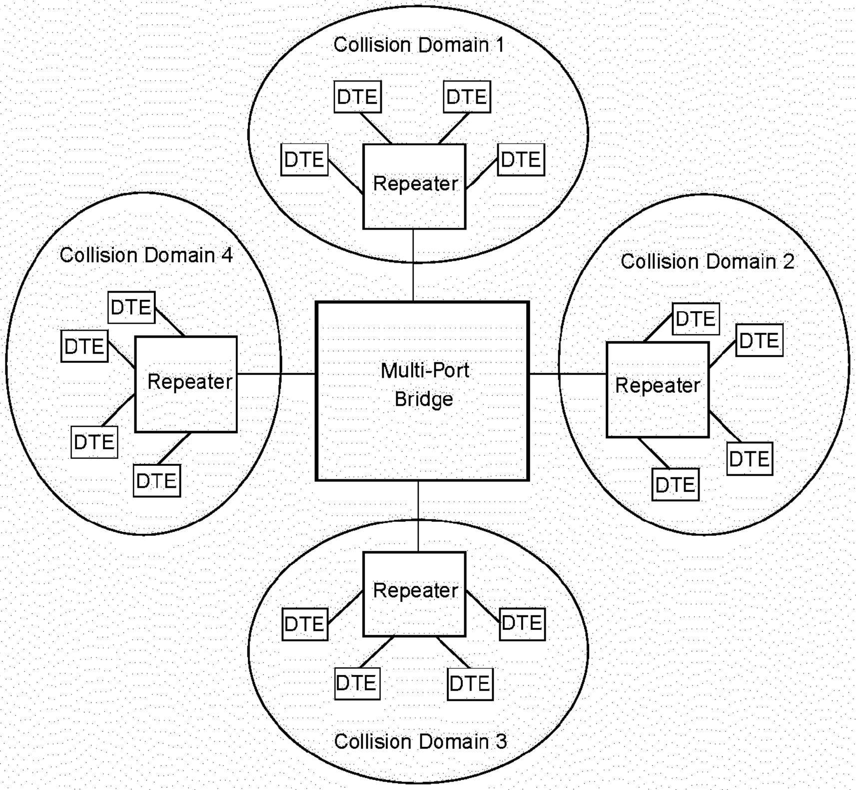


**Figure 29-1—100 Mb/s multiple collision domain topology using multi-port bridge**

Individual collision domains can be linked by single devices (as shown in figures 29-1 and 29-2) or by multiple devices from any of several transmission systems. The design of multiple-collision-domain networks is governed by the rules defining each of the transmission systems incorporated into the design.

The design of shared bandwidth 10 Mb/s collision domains is defined in clause 13; the design of shared bandwidth 100 Mb/s CSMA/CD collision domains is defined in the following subclauses.

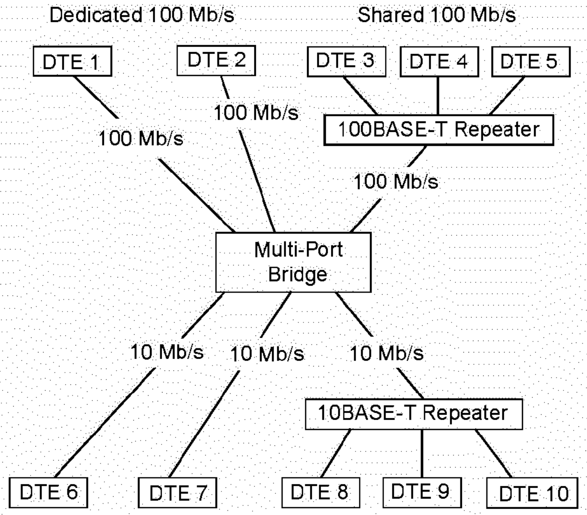


**Figure 29-2—Multiple bandwidth, multiple collision domain topology using multi-port bridge**

### 29.1.1 Single collision domain multi-segment networks

This clause provides information on building 100 Mb/s CSMA/CD multi-segment networks within a single collision domain. The proper operation of a CSMA/CD network requires the physical size and number of repeaters to be limited in order to meet the round-trip propagation delay requirements of 4.2.3.2.3 and 4.4.2.1 and IPG requirements specified in 4.4.2.1.

This clause provides two network models. Transmission System Model 1 is a set of configurations that have been validated under conservative rules and have been qualified as meeting the requirements set forth above. Transmission System Model 2 is a set of calculation aids that allow those configuring a network to test a proposed configuration against a simple set of criteria that allows it to be qualified. Transmission System Model 2 validates an additional broad set of topologies that are fully functional and do not fit within the simpler, but more restrictive rules of Model 1.

The physical size of a CSMA/CD network is limited by the characteristics of individual network components. These characteristics include the following:

a) Media lengths and their associated propagation time delay
b) Delay of repeater units (start-up, steady-state, and end of event)
c) Delay of MAUs and PHYs (start-up, steady-state, and end of event)
d) Interpacket gap shrinkage due to repeater units
e) Delays within the DTE associated with the CSMA/CD access method
f) Collision detect and deassertion times associated with the MAUs and PHYs

Table 29-1 summarizes the delays for 100BASE-T media segments. For more detailed information on the delays associated with individual 100BASE-T components, see

| | |
|---|---|
| MII: | annex 22A |
| 100BASE-T4: | 23.11 |
| 100BASE-TX: | annex 24A |
| 100BASE-FX: | annex 24A |

Repeater: 27.3

Table 29-1—Delays for network media segments Model 1

| Media type | Maximum number of PHYs per segment | Maximum segment length (m) | Maximum medium round-trip delay per segment (BT) |
|---|---|---|---|
| Balanced cable Link Segment 100BASE-T | 2 | 100 | 114 |
| Fiber Link Segment | 2 | 412 | 412 |

### 29.1.2 Repeater usage

Repeaters are the means used to connect segments of a network medium together into a single collision domain. Different physical signaling systems (e.g., 100BASE-T4, 100BASE-TX, 100BASE-FX) can be joined into a common collision domain using repeaters. Bridges can also be used to connect different signaling systems; however, if a bridge is so used, each system connected to the bridge will be a separate collision domain.

Two types of repeaters are defined for 100BASE-T (see clause 27). Class I repeaters are principally used to connect unlike physical signaling systems and have internal delays such that only one Class I repeater can reside within a single collision domain when maximum cable lengths are used (see figure 29-4). Class II repeaters typically provide ports for only one physical signaling system type (e.g., 100BASE-TX but not 100BASE-T4) and have smaller internal delays so that two such repeaters may reside within a given collision domain when maximum cable lengths are used (see figure 29-6). Cable length can be sacrificed to add additional repeaters in a collision domain (see 29.3).

## 29.2 Transmission System Model 1

The following network topology constraints apply to networks using Transmission System Model 1.

a) All balanced cable (copper) segments less than or equal to 100 m each.
b) Fiber segments less than or equal to 412 m each.
c) MII cables for 100BASE-T shall not exceed 0.5 m each. When evaluating system topology, MII cable delays need not be accounted for separately. Delays attributable to the MII are incorporated into DTE and repeater component delays.

## 29.3 Transmission System Model 2

The physical size and number of topological elements in a 100BASE-T network is limited primarily by round-trip collision delay. A network configuration must be validated against collision delay using a network model. Since there are a limited number of topology models for any 100BASE-T collision domain, the modeling process is quite straightforward and can easily be done either manually or with a spreadsheet.

The model proposed here is derived from the one presented in 13.4. Modifications have been made to accommodate adjustments for DTE, repeater, and cable speeds.

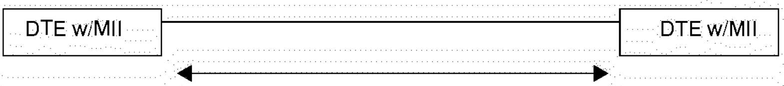


Figure 29-3—Model 1: Two DTEs, no repeater

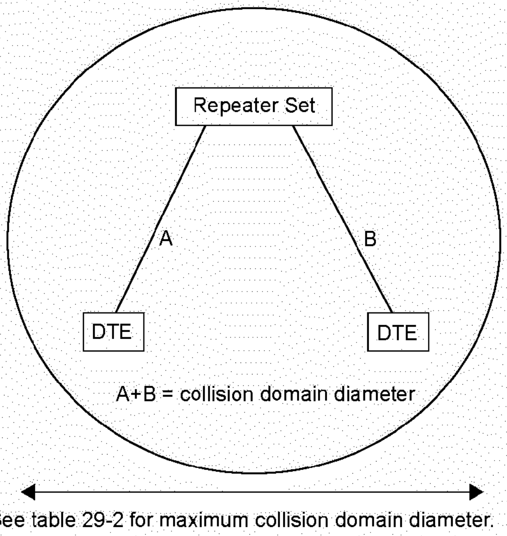


Figure 29-4—Model 1: Single repeater

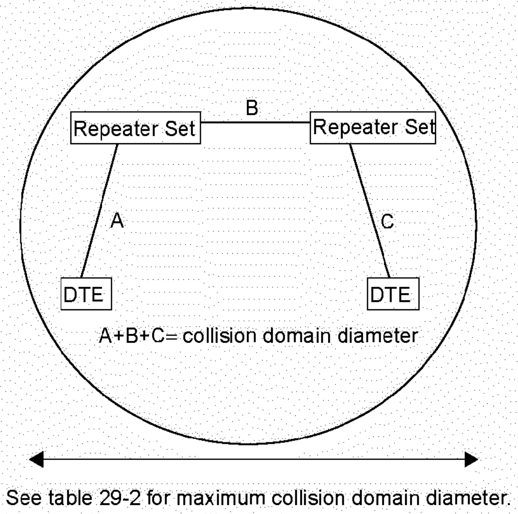


Figure 29-5—System Model 1: Two Class II repeaters

### Table 29-2—Maximum Model 1 collision domain diameter[a]

| Model | Balanced cable (copper) | Fiber | Balanced cable & fiber (T4 and FX) | Balanced cable & fiber (TX and FX) |
|---|---|---|---|---|
| DTE-DTE (see figure 29-3) | 100 | 412 | na | na |
| One Class I repeater (see figure 29-4) | 200 | 272 | 231[b] | 260.8[b] |
| One Class II repeater (see figure 29-4) | 200 | 320 | 304[b,c] | 308.8[b] |
| Two Class II repeaters (see figure 29-5) | 205 | 228 | 236.3[d,c] | 216.2[d] |

[a] In meters, no margin.
[b] Assumes 100 m of balanced cable and one fiber link.
[c] This entry included for completeness. It may be impractical to construct a T4 to FX class II repeater.
[d] Assumes 105 m of balanced cable and one fiber link.

### 29.3.1 Round-trip collision delay

For a network to be valid, it must be possible for any two DTEs on the network to contend for the network at the same time. Each station attempting to transmit must be notified of the contention by the returned "collision" signal within the "collision window" (see 4.1.2.2 and 5.2.2.1.2). Additionally, the maximum length fragment created must contain less than 512 bits after the start-of-frame delimiter (SFD). These requirements limit the physical diameter (maximum distance between DTEs) of a network. The maximum round-trip delay must be qualified between all pairs of DTEs in the network. In practice this means that the qualification must be done between those that, by inspection of the topology, are candidates for the longest delay. The following network modeling methodology is provided to assist that calculation.

#### 29.3.1.1 Worst-case path delay value (PDV) selection

The worst-case path through a network to be validated shall be identified by examination of aggregate DTE delays, cable delays, and repeater delays. The worst case consists of the path between the two DTEs at opposite ends of the network that have the longest round-trip time. Figures 29-6 and 29-7 show schematic representatins of one-repeater and two-repeater paths.

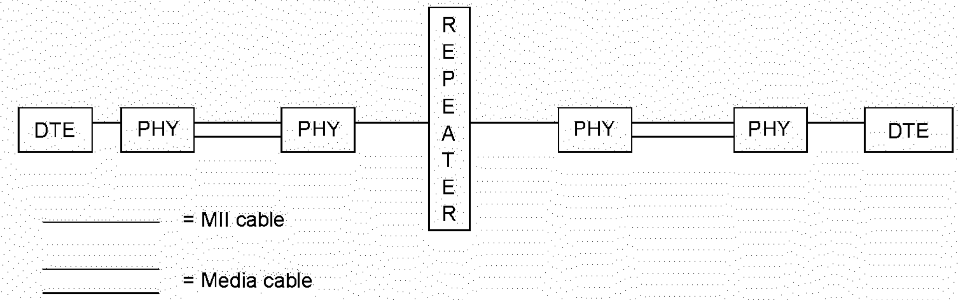


Figure 29-6—System Model 2: Single repeater

#### 29.3.1.2 Worst-case PDV calculation

Once a set of paths is chosen for calculation, each shall be checked for validity against the following formula:

PDV = • link delays (LSDV) + • repeater delays + DTE delays + safety margin

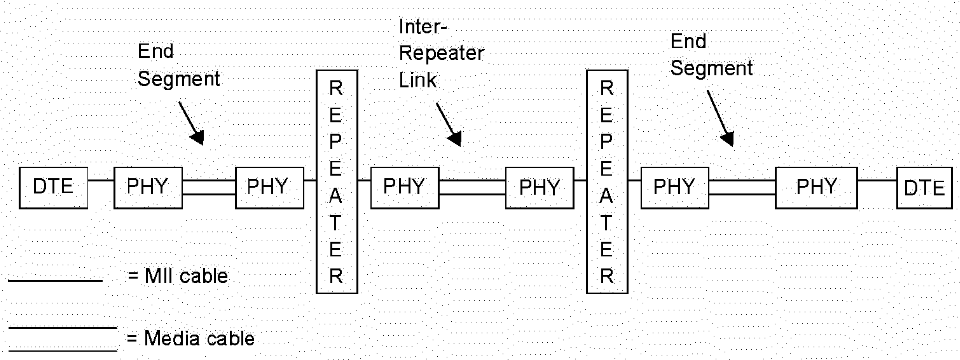


**Figure 29-7—System Model 2-2: Two repeaters**

Values for the formula variables are determined by the following method:

a) Determine the delay for each link segment (Link Segment Delay Value, or LSDV), including inter-repeater links, using the formula

LSDV=2 (for round-trip delay) × segment length × cable delay for this segment

NOTES

1—Length is the sum of the cable lengths between the PHY interfaces at the repeater and the farthest DTE for End Segments plus the sum of the cable lengths between the repeater PHY interfaces for Inter-Repeater Links. All measurements are in meters.

2—Cable delay is the delay specified by the manufacturer or the maximum value for the type of cable used as shown in table 29-3. For this calculation, cable delay must be specified in bit times per meter (BT/m). Table 29-4 can be used to convert values specified relative to the speed of light (%c) or nanoseconds per meter (ns/m).

3—When actual cable lengths or propagation delays are not known, use the Max delay in bit times as specified in table 29-3 for copper cables. Delays for fiber should be calculated, as the value found in table 29-3 will be too large for most applications.

b) Sum together the LSDVs for all segments in the path.
c) Determine the delay for each repeater in the path. If model-specific data are not available from the manufacturer, determine the class of each repeater (I or II) and enter the appropriate default value from table 29-3.
d) MII cables for 100BASE-T shall not exceed 0.5 m each. When evaluating system topology, MII cable delays need not be accounted for separately. Delays attributable to the MII are incorporated into DTE and repeater component delays.
e) Use the DTE delay value shown in table 29-3 unless your equipment manufacturer defines a different value.
f) Decide on appropriate safety margin—0 to 5 bit times—for the PDV calculation. Safety margin is used to provide additional margin to accommodate unanticipated delay elements, such as extra-long connecting cable runs between wall jacks and DTEs. (A safety margin of 4 BT is recommended.)
g) Insert the values obtained through the calculations above into the following formula to calculate the PDV. (Some configurations may not use all the elements of the formula.)

PDV = • link delays (LSDV) + • repeater delays + DTE delay + safety margin

h) If the PDV is less than 512, the path is qualified in terms of worst-case delay.
i) Late collisions and/or CRC errors are indicators that path delays exceed 512 BT.

### Table 29-3—Network component delays, Transmission System Model 2

| Component | Round-trip delay in bit times per meter | Maximum round-trip delay in bit times |
|---|---|---|
| Two TX/FX DTEs | | 100 |
| Two T4 DTEs | | 138 |
| One T4 and one TX/FX DTE[a] | | 127 |
| Cat 3 cable segment | 1.14 | 114 (100 m) |
| Cat 4 cable segment | 1.14 | 114 (100 m) |
| Cat 5 cable segment | 1.112 | 111.2 (100 m) |
| STP cable segment | 1.112 | 111.2 (100 m) |
| Fiber optic cable segment | 1.0 | 412 (412 m) |
| Class I repeater | | 140 |
| Class II repeater with all ports TX/FX | | 92 |
| Class II repeater with any port T4 | | 67 |

[a] Worst-case values are used (TX/FX values for MAC transmit start and MDI input to collision detect; T4 value for MDI input to MDI output).

### Table 29-4—Conversion table for cable delays

| Speed relative to c | ns/m | BT/m |
|---|---|---|
| 0.4 | 8.34 | 0.834 |
| 0.5 | 6.67 | 0.667 |
| 0.51 | 6.54 | 0.654 |
| 0.52 | 6.41 | 0.641 |
| 0.53 | 6.29 | 0.629 |
| 0.54 | 6.18 | 0.618 |
| 0.55 | 6.06 | 0.606 |
| 0.56 | 5.96 | 0.596 |
| 0.57 | 5.85 | 0.585 |
| 0.58 | 5.75 | 0.575 |
| 0.5852 | 5.70 | 0.570 |
| 0.59 | 5.65 | 0.565 |
| 0.6 | 5.56 | 0.556 |
| 0.61 | 5.47 | 0.547 |
| 0.62 | 5.38 | 0.538 |
| 0.63 | 5.29 | 0.529 |
| 0.64 | 5.21 | 0.521 |
| 0.65 | 5.13 | 0.513 |
| 0.654 | 5.10 | 0.510 |
| 0.66 | 5.05 | 0.505 |
| 0.666 | 5.01 | 0.501 |
| 0.67 | 4.98 | 0.498 |
| 0.68 | 4.91 | 0.491 |
| 0.69 | 4.83 | 0.483 |
| 0.7 | 4.77 | 0.477 |
| 0.8 | 4.17 | 0.417 |
| 0.9 | 3.71 | 0.371 |

# 30. Layer Management for 10 Mb/s and 100 Mb/s

## 30.1 Overview

This clause provides the Layer Management specification for DTEs, repeaters, and MAUs based on the CSMA/CD access method. The clause is produced from the ISO framework additions to clause 5, Layer Management; clause 19, Repeater Management; and clause 20, MAU Management. It incorporates additions to the objects, attributes, and behaviors to support 100 Mb/s CSMA/CD.

The layout of this clause takes the same form as 5.1, 5.2, and clauses 19 and 20, although with equivalent subclauses grouped together. It identifies a common management model and framework applicable to IEEE 802.3 managed elements, and it identifies those elements and defines their managed objects, attributes, and behaviors in a protocol-independent language. It also includes a formal GDMO definition of the protocol encodings for CMIP and ISO/IEC 15802-2: 1995 [IEEE 802.1B].

NOTE—The arcs (that is, object identifier values) defined in annex 30A, the formal GDMO definitions, deprecate the arcs previously defined in Annexes D1 (Layer Management), D2 (Repeater Management), and D3 (MAU Management). See IEEE Std 802.1F-1993, annex C.4.

This clause provides the Layer Management specification for DTEs, repeaters, and MAUs based on the CSMA/CD access method. It defines facilities comprised of a set of statistics and actions needed to provide IEEE 802.3 Management services. The information in this clause should be used in conjunction with the Procedural Model defined in 4.2.7–4.2.10. The Procedural Model provides a formal description of the relationship between the CSMA/CD Layer Entities and the Layer Management facilities.

This management specification has been developed in accordance with the OSI management architecture as specified in the ISO Management Framework document, ISO/IEC 7498-4: 1989. It is independent of any particular management application or management protocol.

The management facilities defined in this standard may be accessed both locally and remotely. Thus, the Layer Management specification provides facilities that can be accessed from within a station or can be accessed remotely by means of a peer-management protocol operating between application entities.

In CSMA/CD no peer management facilities are necessary for initiating or terminating normal protocol operations or for handling abnormal protocol conditions. The monitoring of these activities is done by the carrier sense and collision detection mechanisms. Since these activities are necessary for normal operation of the protocol, they are not considered to be a function of Layer Management and are, therefore, not discussed in this clause.

Implementation of part or all of 10 Mb/s and 100 Mb/s Management is not a requirement for conformance to clauses 4, 7, 9, 22, 23, 24, 25, 26, 27, or 28.

The intent of this standard is to furnish a management specification that can be used by the wide variety of different devices that may be attached to a network specified by ISO/IEC 8802-3. Thus, a comprehensive list of management facilities is provided.

The improper use of some of the facilities described in this clause may cause serious disruption of the network. In accordance with ISO management architecture, any necessary security provisions should be provided by the Agent in the Local System Environment. This can be in the form of specific security features or in the form of security features provided by the peer communication facilities.

### 30.1.1 Scope

This clause includes selections from clauses 5, 19, and 20. It is intended to be an entirely equivalent specification for the management of 10 Mb/s DTEs, 10 Mb/s baseband repeater units, and 10 Mb/s integrated MAUs. It also includes the additions for management of 100 Mb/s DTEs, repeater units, embedded MAUs, and external PHYs connected with the MII. Implementations of management for 10 Mb/s DTEs, repeater units, and embedded MAUs should follow the requirements of this clause (e.g., a 10 Mb/s implementation should incorporate the attributes to indicate that it is not capable of 100 Mb/s operation).

This clause defines a set of mechanisms that enable management of ISO/IEC 8802-3 10 Mb/s and 100 Mb/s DTEs, baseband repeater units, and integrated Medium Attachment Units (MAUs). In addition, for ports without integral MAUs, attributes are provided for characteristics observable from the AUI of the connected DTE or repeater. Direct management of AUI MAUs that are external to their respective DTEs or repeaters is beyond the scope of this standard. The managed objects within this standard are defined in terms of their behaviour, attributes, actions, notifications, and packages in accordance with IEEE 802.1 and ISO standards for network management. Managed objects are grouped into mandatory and optional packages.

This specification is defined to be independent of any particular management application or management protocol. The means by which the managed objects defined in this standard are accessed is beyond the scope of this standard.

### 30.1.2 Relationship to objects in IEEE Std 802.1F-1993

The following managed object classes, if supported by an implementation, shall be as specified in IEEE Std 802.1F-1993: ResourceTypeID, EWMAMetricMonitor.

**oResourceTypeID**

This object class is mandatory and shall be implemented as defined in IEEE Std 802.1F-1993. This object is bound to oMAC-Entity, oRepeater, and oMAU as defined by the NAMEBINDINGs in 30A.8.1. Note that the binding to oMAU is mandatory only when MII is present. The Entity Relationship Diagram, figure 30-3, shows these bindings pictorially.

**oEWMAMetricMonitor**

This object class is optional. When implemented, it shall be implemented as defined in IEEE Std 802.1F-1993, subject to the specific requirements described below. This object is bound to system as defined by the NAMEBINDINGs in 30A.1.1, 30A.3.1, and 30A.2.1.

Implementations of IEEE 802.3 Management that support the oEWMAMetricMonitor managed object class are required to support values of granularity period as small as one second. Implementations are required to support at least one sequence of low and high thresholds. The granularity period may be set to equal to the moving time period as a minimal conformant implementation.

### 30.1.3 Systems management overview

Within the ISO Open Systems Interconnection (OSI) architecture, the need to handle the special problems of initializing, terminating, and monitoring ongoing activities and assisting in their operations, as well as handling abnormal conditions, is recognized. These needs are collectively addressed by the systems management component of the OSI architecture.

A management protocol is required for the exchange of information between systems on a network. This management standard is independent of any particular management protocol.

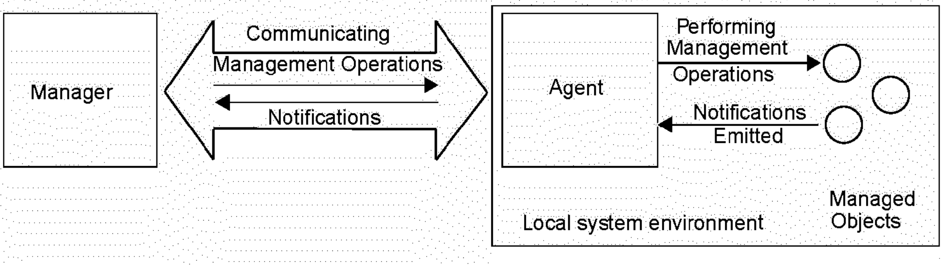
This management standard, in conjunction with the management standards of other layers, provides the means to perform various management functions. IEEE 802.3 Management collects information needed from the MAC and Physical Layers and the devices defined in IEEE 802.3. It also provides a means to exercise control over those elements.

The relationship between the various management entities and the layer entities according to the ISO model is shown in figure 30-1.

### 30.1.4 Management model

This standard describes management of DTEs, repeaters, and integrated MAUs in terms of a general model of management of resources within the open systems environment. The model, which is described in ISO/IEC 10040: 1992, is briefly summarized here.

Management is viewed as a distributed application modeled as a set of interacting management processes. These processes are executed by systems within the open environment. A managing system executes a managing process that invokes management operations. A managed system executes a process that is receptive to these management operations and provides an interface to the resources to be managed. A managed object is the abstraction of a resource that represents its properties as seen by (and for the purpose of) management. Managed objects respond to a defined set of management operations. Managed objects are also capable of emitting a defined set of notifications. This interaction of processes is shown in figure 30-1.



NOTE—Figure 1 of ISO/IEC 10040 has been reproduced with the permission of ISO. Copies of the complete standard may be obtained from the International Organization for Standardization, Case Postale 56, 1 rue de Varembé, CH-1211, Genève 20, Switzerland/Suisse.

**Figure 30-1—Interaction between manager, agent, and objects**

A managed object is a management view of a resource. The resource may be a logical construct, function, physical device, or anything subject to management. Managed objects are defined in terms of four types of elements:

a) *Attributes*. Data-like properties (as seen by management) of a managed object.
b) *Actions*. Operations that a managing process may perform on an object or its attributes.
c) *Notifications*. Unsolicited reports of events that may be generated by an object.
d) *Behaviour*. The way in which managed objects, attributes, and actions interact with the actual resources they model and with each other.

The above items are defined in 30.3, 30.4, 30.5, and 30.6 of this clause in terms of the template requirements of ISO/IEC 10165-4: 1991.

Some of the functions and resources within 802.3 devices are appropriate targets for management. They have been identified by specifying managed objects that provide a management view of the functions or resources. Within this general model, the 802.3 device is viewed as a managed device. It performs functions as defined by the applicable standard for such a device. Managed objects providing a view of those functions and resources appropriate to the management of the device are specified. The purpose of this standard is to define the object classes associated with the devices in terms of their attributes, operations, notifications, and behaviour.

## 30.2 Managed objects

### 30.2.1 Introduction

This clause identifies the Managed Object classes for IEEE 802.3 components within a managed system. It also identifies which managed objects and packages are applicable to which components.

All counters defined in this specification are assumed to be wraparound counters. Wraparound counters are those that automatically go from their maximum value (or final value) to zero and continue to operate. These unsigned counters do not provide for any explicit means to return them to their minimum (zero), i.e., reset. Because of their nature, wraparound counters should be read frequently enough to avoid loss of information. Counters in 30.3, 30.4, 30.5 and 30.6 that have maximum increment rates specified for 10 Mb/s operation, and are appropriate to 100 Mb/s operation, have ten times the stated maximum increment rate for 100 Mb/s operation unless otherwise indicated.

### 30.2.2 Overview of managed objects

Managed objects provide a means to

- Identify a resource
- Control a resource
- Monitor a resource

### 30.2.2.1 Text description of managed objects

In case of conflict, the formal behaviour definitions in 30.3, 30.4, 30.5, and 30.6 take precedence over the text descriptions in this subclause.

| | |
|---|---|
| **oMACEntity** | The top-most managed object class of the DTE portion of the containment tree shown in figure 30-3. Note that this managed object class may be contained within another superior managed object class. Such containment is expected, but is outside the scope of this standard. |
| **oPHYEntity** | Contained within oMACEntity. Many instances of oPHYEntity may coexist within one instance of oMACEntity; however, only one PHY may be active for data transfer to and from the MAC at any one time. oPHYEntity is the managed object that contains the MAU managed object in a DTE. |
| **oRepeater** | The top-most managed object class of the repeater portion of the containment tree shown in figure 30-3. Note that this managed object class may be contained within another superior managed object class. Such containment is expected, but is outside the scope of this standard. |

| | |
|---|---|
| **oRepeaterMonitor** | A managed object class called out by IEEE Std 802.1F-1993. See 30.1.2, oEWMAMetricMonitor. |
| **oGroup** | The group managed object class is a view of a collection of repeater ports. |
| **oRepeaterPort** | The repeater port managed object class provides a view of the functional link between the data transfer service and a single PMA. The attributes associated with repeater port deal with the monitoring of traffic being handled by the repeater from the port and control of the operation of the port. The Port Enable/Disable function as reported by portAdminState is preserved across events involving loss of power. The oRepeaterPort managed object contains the MAU managed object in a repeater set.<br>NOTE—Attachment to nonstandard PMAs is outside the scope of this standard. |
| **oMAU** | The managed object of that portion of the containment tree shown in figure 30-3. The attributes, notifications, and actions defined in this clause are contained within the MAU managed object. Neither counter values nor the value of MAUAdminState is required to be preserved across events involving the loss of power. |
| **oAutoNegotiation** | The managed object of that portion of the containment tree shown in figure 30-3. The attributes, notifications, and actions defined in this clause are contained within the MAU managed object. |
| **oResourceTypeID** | A managed object class called out by IEEE Std 802.1F-1993. It is used within this clause to identify manufacturer, product, and revision of managed components that implement functions and interfaces defined within IEEE 802.3. The clause 22 MII specifies two registers to carry PHY Identifier (22.2.4.3.1), which provides succinct information sufficient to support oResourceTypeID. |

### 30.2.2.2 Functions to support management

Functions are defined in clauses 5, 7, 22, 23, 24, 25, 26, 27, and 28 both to facilitate unmanaged operation and managed operation. The functions in these clauses that facilitate managed operation are referenced from the text of this management clause.

#### 30.2.2.2.1 DTE MAC sublayer functions

For DTE MACs, with regard to reception-related error statistics a hierarchical order has been established such that when multiple error statuses can be associated with one frame, only one status is returned to the LLC. This hierarchy in descending order is as follows:

- frameTooLong
- alignmentError
- frameCheckError
- lengthError

The counters are primarily incremented based on the status returned to the LLC; therefore, the hierarchical order of the counters is determined by the order of the status. Frame fragments are not included in any of the statistics unless otherwise stated. In implementing any of the specified actions, receptions and transmissions that are in progress are completed before the action takes effect.

### 30.2.2.2.2 Repeater functions

The Repeater Port Object class contains seven functions which are defined in this clause and are used to collect statistics on the activity received by the port. The relationship of the functions to the repeater port and to the port attributes is shown in figure 30-2.

**Activity Timing function**

The Activity Timing function measures the duration of the assertion of the CarrierEvent signal. This duration value must be adjusted by removing the value of Carrier Recovery Time (see 9.5.6.5) to obtain the true duration of activity on the network. The output of the Activity Timing function is the ActivityDuration value, which represents the duration of the CarrierEvent signal as expressed in units of bit times.

**Carrier Event function**

The Carrier Event function asserts the CarrierEvent signal when the repeater exits the IDLE state (see figure 9-2) and the port has been determined to be port N. It de-asserts the CarrierEvent signal when, for a duration of at least Carrier Recovery Time (see 9.5.6.5), both the DataIn(N) variable has the value II and the CollIn(N) variable has the value –SQE. The value N is the port assigned at the time of transition from the IDLE state.

**Collision Event function**

The Collision Event function asserts the CollisionEvent signal when the CollIn(X) variable has the value SQE. The CollisionEvent signal remains asserted until the assertion of any CarrierEvent signal due to the reception of the following event.

**Cyclic Redundancy Check function**

The Cyclic Redundancy Check function verifies that the sequence of octets output by the Framing function contains a valid Frame Check Sequence Field. The Frame Check Sequence Field is the last four octets received from the output of the Framing function. The algorithm for generating an FCS from the octet stream is specified in 3.2.8. If the FCS generated according to this algorithm is not the same as the last four octets received from the Framing function, then the FCSError signal is asserted. The FCSError signal is cleared upon the assertion of the CarrierEvent signal due to the reception of the following event.

**Framing function**

The Framing function recognizes the boundaries of an incoming frame by monitoring the CarrierEvent signal and the decoded data stream. Data bits are accepted while the CarrierEvent signal is asserted. The framing function strips preamble and start-of-frame delimiter from the received data stream. The remaining bits are aligned along octet boundaries. If there is not an integral number of octets, then FramingError shall be asserted. The FramingError signal is cleared upon the assertion of the CarrierEvent signal due to the reception of the following event.

**Octet Counting function**

The Octet Counting function counts the number of complete octets received from the output of the framing function. The output of the octet counting function is the OctetCount value. The OctetCount value is reset to zero upon the assertion of the CarrierEvent signal due to the reception of the following event.

**Source Address function**

The Source Address function extracts octets from the stream output by the framing function. The seventh through twelfth octets shall be extracted from the octet stream and output as the SourceAddress variable. The SourceAddress variable is set to an invalid state upon the assertion of the CarrierEvent signal due to the reception of the following event.

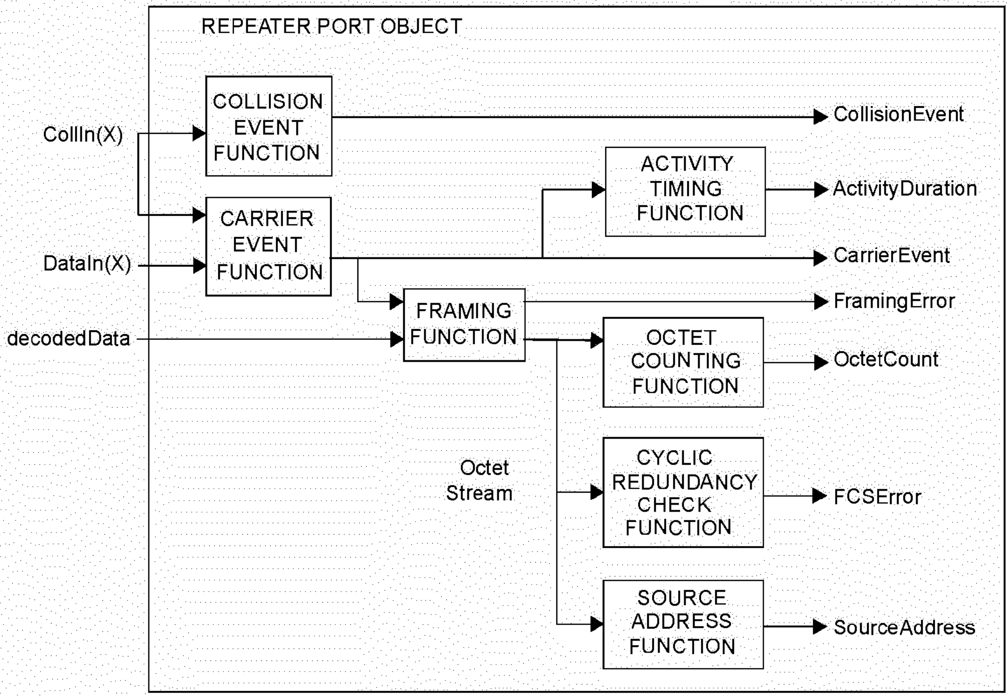


Figure 30-2—Functions relationship

### 30.2.3 Containment

A containment relationship is a structuring relationship for managed objects in which the existence of a managed object is dependent on the existence of a containing managed object. The contained managed object is said to be the subordinate managed object, and the containing managed object the superior managed object. The containment relationship is used for naming managed objects. The local containment relationships among object classes are depicted in the entity relationship diagram, figure 30-3. This figure also shows the names, naming attributes, and data attributes of the object classes as well as whether a particular containment relationship is one-to-one or one-to-many. For further requirements on this topic, see IEEE Std 802.1F-1993.

MAU management is only valid in a system that provides management at the next higher containment level, that is, either a DTE or repeater with management.

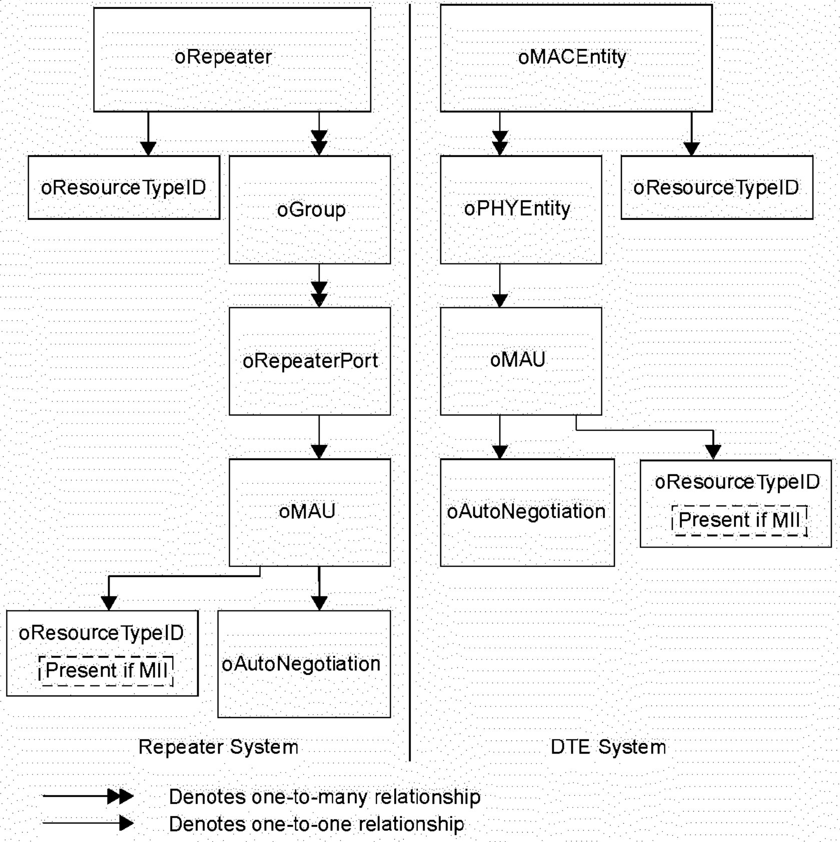


**Figure 30-3—10/100 Mb/s entity relationship diagram**

### 30.2.4 Naming

The name of an individual managed object is hierarchically defined within a managed system. For example, in the context of repeater management, a repeater port might be identified as "repeater 3, group 01, port 13," that is, port 13 of group 01 of a repeater with repeaterID 3 within the managed system.

In the case of MAU management, this will present itself in one of the two forms that are appropriate for a MAU's use, that is, as associated with a CSMA/CD interface of a DTE or with a particular port of a managed repeater. For example, a MAU could be identified as "repeater 3, group 01, port 13, MAU 1" or, that is, the MAU associated with port 13 of group 01 of a repeater with repeaterID 3 within the managed system. Examples of this are represented in the relationship of the naming attributes in the entity relationship diagram, figure 30-3.

### 30.2.5 Capabilities

This standard makes use of the concept of *packages* as defined in ISO/IEC 10165-4: 1992 as a means of grouping behaviour, attributes, actions, and notifications within a managed object class definition. Packages may either be mandatory or conditional, that is to say, present if a given condition is true. Within this standard, *capabilities* are defined, each of which corresponds to a set of packages, which are components of a number of managed object class definitions and which share the same condition for presence. Implementation of the appropriate basic and the mandatory packages is the minimum requirement for claiming conformance to IEEE 802.3 10 Mb/s and 100 Mb/s Management. Implementation of an entire optional capability is required in order to claim conformance to that capability. The capabilities and packages for 10 Mb/s and 100 Mb/s Management are specified in table 30-1 (broken into tables 30-1a through 30-1d for pagination).

DTE Management has two packages that are required for management at the minimum conformance configuration—the Basic Package and the Mandatory Package. For systems that include multiple PHY entities per MAC entity, and implement the Multiple PHY Package to manage the selection of the active PHY, the optional Recommended Package shall be implemented.

For managed MAUs, the Basic Package is mandatory; all other packages are optional. For a managed MAU to be conformant to this standard, it shall fully implement the Basic Package. For a MAU to be conformant to an optional package, it shall implement that entire package. While nonconformant (reference aMAUType "other") MAUs may utilize some or all of this clause to specify their management, conformance to this clause requires both a conformant MAU and conformant management. MAU Management is optional with respect to all other CSMA/CD Management. If an MII is present, then the conditional MII Capability must be implemented. This provides the means to identify the vendor and type of the externally connected device.

There are two distinct aspects of Repeater Management.

The first aspect provides the means to monitor and control the functions of a repeater. These functions include, but are not limited to identifying a repeater, testing and initializing a repeater, and enabling/disabling a port. This is encompassed by the mandatory Basic Control Capability.

The second aspect provides the means to monitor traffic from attached segments, and to measure traffic sourced by DTEs connected to these segments. This is done by gathering statistics on packets that enter a repeater and maintaining those statistics on a per-port basis. This is encompassed by the optional Performance Monitor Capability. The optional Address Tracking Capability provides the means to identify existence and movement of attached DTEs by their MAC addresses.

If link Auto-Negotiation is present and managed, the Auto-Negotiation managed object class shall be implemented in its entirety. All attributes and actions are mandatory.

The 100 Mb/s Monitor Capability provides additional attributes that relate to 100 Mb/s operation only. These attributes are provided to complement the counter attributes of the optional packages and capabilities that apply to 10 Mb/s and mixed 10 and 100 Mb/s implementations. It is expected that when the 100 Mb/s Monitor Capability is implemented, the appropriate complementary counter packages and capabilities are also implemented.

## Table 30-1a—Capabilities

| | | | DTE | | | | | | | | Repeater | | | | MAU | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Basic Package (Mandatory) | Mandatory Package (Mandatory) | Recommended Package (Optional) | Optional Package (Optional) | Array Package (Optional) | Excessive Deferral Package (Optional) | Multiple PHY Package (Optional) | 100 Mb/s Monitor Capability (Optional) | Basic Control Capability (Mandatory) | Performance Monitor Capability (Optional) | Address Tracking Capability (Optional) | 100 Mb/s Monitor Capability (Optional) | Basic Package (Mandatory) | MAU Control Package (Optional) | Media Loss Tracking Package (Conditional) | Broadband DTE MAU Package (Conditional) | MII Capability (Conditional) | 100 Mb/s Monitor Capability (Optional) | Auto-Negotiation Package (Mandatory) |
| **oResourceTypeID managed object** | | | | | | | | | | | | | | | | | | | | | |
| aResourceTypeIDName | ATTRIBUTE | GET | X | | | | | | | | X | | | | | | | | X | | |
| aResourceInfo | ATTRIBUTE | GET | X | | | | | | | | X | | | | | | | | X | | |
| **oMACEntity managed object class** | | | | | | | | | | | | | | | | | | | | | |
| aMACID | ATTRIBUTE | GET | X | | | | | | | | | | | | | | | | | | |
| aFramesTransmittedOK | ATTRIBUTE | GET | | X | | | | | | | | | | | | | | | | | |
| aSingleCollisionFrames | ATTRIBUTE | GET | | X | | | | | | | | | | | | | | | | | |
| aMultipleCollisionFrames | ATTRIBUTE | GET | | X | | | | | | | | | | | | | | | | | |
| aFramesReceivedOK | ATTRIBUTE | GET | | X | | | | | | | | | | | | | | | | | |
| aFrameCheckSequenceErrors | ATTRIBUTE | GET | | X | | | | | | | | | | | | | | | | | |
| aAlignmentErrors | ATTRIBUTE | GET | | X | | | | | | | | | | | | | | | | | |
| aOctetsTransmittedOK | ATTRIBUTE | GET | | | X | | | | | | | | | | | | | | | | |
| aFramesWithDeferredXmissions | ATTRIBUTE | GET | | | X | | | | | | | | | | | | | | | | |
| aLateCollisions | ATTRIBUTE | GET | | | X | | | | | | | | | | | | | | | | |
| aFramesAbortedDueToXSColls | ATTRIBUTE | GET | | | X | | | | | | | | | | | | | | | | |
| aFramesLostDueToIntMACXmitError | ATTRIBUTE | GET | | | X | | | | | | | | | | | | | | | | |
| aCarrierSenseErrors | ATTRIBUTE | GET | | | X | | | | | | | | | | | | | | | | |
| aOctetsReceivedOK | ATTRIBUTE | GET | | | X | | | | | | | | | | | | | | | | |
| aFramesLostDueToIntMACRcvError | ATTRIBUTE | GET | | | X | | | | | | | | | | | | | | | | |
| aPromiscuousStatus | ATTRIBUTE | GET-SET | | | X | | | | | | | | | | | | | | | | |
| aReadMulticastAddressList | ATTRIBUTE | GET | | | X | | | | | | | | | | | | | | | | |
| aMulticastFramesXmittedOK | ATTRIBUTE | GET | | | | X | | | | | | | | | | | | | | | |
| aBroadcastFramesXmittedOK | ATTRIBUTE | GET | | | | X | | | | | | | | | | | | | | | |
| aFramesWithExcessiveDeferral | ATTRIBUTE | GET | | | | | | X | | | | | | | | | | | | | |
| aMulticastFramesReceivedOK | ATTRIBUTE | GET | | | | X | | | | | | | | | | | | | | | |
| aBroadcastFramesReceivedOK | ATTRIBUTE | GET | | | | X | | | | | | | | | | | | | | | |
| aInRangeLengthErrors | ATTRIBUTE | GET | | | | X | | | | | | | | | | | | | | | |
| aOutOfRangeLengthField | ATTRIBUTE | GET | | | | X | | | | | | | | | | | | | | | |
| aFrameTooLongErrors | ATTRIBUTE | GET | | | | X | | | | | | | | | | | | | | | |
| aMACEnableStatus | ATTRIBUTE | GET-SET | | | | X | | | | | | | | | | | | | | | |

## Table 30-1b—Capabilities

| | | | DTE | | | | | | | | Repeater | | | | MAU | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Basic Package (Mandatory) | Mandatory Package (Mandatory) | Recommended Package (Optional) | Optional Package (Optional) | Array Package (Optional) | Excessive Deferral Package (Optional) | Multiple PHY Package (Optional) | 100 Mb/s Monitor Capability (Optional) | Basic Control Capability (Mandatory) | Performance Monitor Capability (Optional) | Address Tracking Capability (Optional) | 100 Mb/s Monitor Capability (Optional) | Basic Package (Mandatory) | MAU Control Package (Optional) | Media Loss Tracking Package (Conditional) | Broadband DTE MAU Package (Conditional) | MII Capability (Conditional) | 100 Mb/s Monitor Capability (Optional) | Auto-Negotiation Package (Mandatory) |
| **oMACEntity managed object class (con'd.)** | | | | | | | | | | | | | | | | | | | | | |
| aTransmitEnableStatus | ATTRIBUTE | GET-SET | | | | X | | | | | | | | | | | | | | | |
| aMulticastReceiveStatus | ATTRIBUTE | GET-SET | | | | X | | | | | | | | | | | | | | | |
| aReadWriteMACAddress | ATTRIBUTE | GET-SET | | | | X | | | | | | | | | | | | | | | |
| aCollisionFrames | ATTRIBUTE | GET | | | | | X | | | | | | | | | | | | | | |
| acInitializeMAC | ACTION | | X | | | | | | | | | | | | | | | | | | |
| acAddGroupAddress | ACTION | | | | X | | | | | | | | | | | | | | | | |
| acDeleteGroupAddress | ACTION | | | | X | | | | | | | | | | | | | | | | |
| acExecuteSelfTest | ACTION | | | | | X | | | | | | | | | | | | | | | |
| **oPHYEntity managed object class** | | | | | | | | | | | | | | | | | | | | | |
| aPHYID | ATTRIBUTE | GET | X | | | | | | | | | | | | | | | | | | |
| aPHYType | ATTRIBUTE | GET | X | | | | | | | | | | | | | | | | | | |
| aPHYTypeList | ATTRIBUTE | GET | X | | | | | | | | | | | | | | | | | | |
| aSQETestErrors | ATTRIBUTE | GET | | | X | | | | | | | | | | | | | | | | |
| aSymbolErrorDuringCarrier | ATTRIBUTE | GET | | | | | | | | X | | | | | | | | | | | |
| aMIIDetect | ATTRIBUTE | GET | X | | | | | | | | | | | | | | | | | | |
| aPHYAdminState | ATTRIBUTE | GET | X | | | | | | | | | | | | | | | | | | |
| acPHYAdminControl | ACTION | | | | | | | | X | | | | | | | | | | | | |
| **oRepeater managed object class** | | | | | | | | | | | | | | | | | | | | | |
| aRepeaterID | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aRepeaterType | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aRepeaterGroupCapacity | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aGroupMap | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aRepeaterHealthState | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aRepeaterHealthText | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aRepeaterHealthData | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aTransmitCollisions | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| acResetRepeater | ACTION | | | | | | | | | | X | | | | | | | | | | |
| acExecuteNonDisruptiveSelfTest | ACTION | | | | | | | | | | X | | | | | | | | | | |
| nRepeaterHealth | NOTIFICATION | | | | | | | | | | X | | | | | | | | | | |
| nRepeaterReset | NOTIFICATION | | | | | | | | | | X | | | | | | | | | | |
| nGroupMapChange | NOTIFICATION | | | | | | | | | | X | | | | | | | | | | |

## Table 30-1c—Capabilities

| | | | DTE | | | | | | | | Repeater | | | | MAU | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Basic Package (Mandatory) | Mandatory Package | Recommended Package (Optional) | Optional Package (Optional) | Array Package (Optional) | Excessive Deferral Package (Optional) | Multiple PHY Package (Optional) | 100 Mb/s Monitor Capability (Optional) | Basic Control Capability (Mandatory) | Performance Monitor Capability (Optional) | Address Tracking Capability (Optional) | 100 Mb/s Monitor Capability (Optional) | Basic Package (Mandatory) | MAU Control Package (Optional) | Media Loss Tracking Package (Conditional) | Broadband DTE MAU Package (Conditional) | MII Capability (Conditional) | 100 Mb/s Monitor Capability (Optional) | Auto-Negotiation Package (Mandatory) |
| **oGroup managed object class** | | | | | | | | | | | | | | | | | | | | | |
| aGroupID | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aGroupPortCapacity | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aPortMap | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| nPortMapChange | NOTIFICATION | | | | | | | | | | X | | | | | | | | | | |
| **oRepeaterPort managed object class** | | | | | | | | | | | | | | | | | | | | | |
| aPortID | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aPortAdminState | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aAutoPartitionState | ATTRIBUTE | GET | | | | | | | | | X | | | | | | | | | | |
| aReadableFrames | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aReadableOctets | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aFrameCheckSequenceErrors | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aAlignmentErrors | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aFramesTooLong | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aShortEvents | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aRunts | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aCollisions | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aLateEvents | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aVeryLongEvents | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aDataRateMismatches | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aAutoPartitions | ATTRIBUTE | GET | | | | | | | | | | X | | | | | | | | | |
| aIsolates | ATTRIBUTE | GET | | | | | | | | | | | | X | | | | | | | |
| aSymbolErrorDuringPacket | ATTRIBUTE | GET | | | | | | | | | | | | X | | | | | | | |
| aLastSourceAddress | ATTRIBUTE | GET | | | | | | | | | | | X | | | | | | | | |
| aSourceAddressChanges | ATTRIBUTE | GET | | | | | | | | | | | X | | | | | | | | |
| acPortAdminControl | ACTION | | | | | | | | | | X | | | | | | | | | | |

## Table 30-1d—Capabilities

| | | | DTE | | | | | | | | Repeater | | | | MAU | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Basic Package (Mandatory) | Mandatory Package | Recommended Package (Optional) | Optional Package (Optional) | Array Package (Optional) | Excessive Deferral Package (Optional) | Multiple PHY Package (Optional) | 100 Mb/s Monitor Capability (Optional) | Basic Control Capability (Mandatory) | Performance Monitor Capability (Optional) | Address Tracking Capability (Optional) | 100 Mb/s Monitor Capability (Optional) | Basic Package (Mandatory) | MAU Control Package (Optional) | Media Loss Tracking Package (Conditional) | Broadband DTE MAU Package (Conditional) | MII Capability (Conditional) | 100 Mb/s Monitor Capability (Optional) | Auto-Negotiation Package (Mandatory) |
| **oMAU managed object class** | | | | | | | | | | | | | | | | | | | | | |
| aMAUID | ATTRIBUTE | GET | | | | | | | | | | | | | X | | | | | | |
| aMAUType | ATTRIBUTE | GET-SET | | | | | | | | | | | | | X | | | | | | |
| aMAUTypeList | ATTRIBUTE | GET | | | | | | | | | | | | | X | | | | | | |
| aMediaAvailable | ATTRIBUTE | GET | | | | | | | | | | | | | X | | | | | | |
| aLoseMediaCounter | ATTRIBUTE | GET | | | | | | | | | | | | | | | X | | | | |
| aJabber | ATTRIBUTE | GET | | | | | | | | | | | | | X | | | | | | |
| aMAUAdminState | ATTRIBUTE | GET | | | | | | | | | | | | | X | | | | | | |
| aBbMAUXmitRcvSplitType | ATTRIBUTE | GET | | | | | | | | | | | | | | | | X | | | |
| aBroadbandFrequencies | ATTRIBUTE | GET | | | | | | | | | | | | | | | | X | | | |
| aFalseCarriers | ATTRIBUTE | GET | | | | | | | | | | | | | | | | | | X | |
| acResetMAU | ACTION | | | | | | | | | | | | | | | X | | | | | |
| acMAUAdminControl | ACTION | | | | | | | | | | | | | | | X | | | | | |
| nJabber | NOTIFICATION | | | | | | | | | | | | | | X | | | | | | |
| **oAuto-Negotiation managed object class** | | | | | | | | | | | | | | | | | | | | | |
| aAutoNegID | ATTRIBUTE | GET | | | | | | | | | | | | | | | | | | | X |
| aAutoNegAdminState | ATTRIBUTE | GET | | | | | | | | | | | | | | | | | | | X |
| aAutoNegRemoteSignaling | ATTRIBUTE | GET | | | | | | | | | | | | | | | | | | | X |
| aAutoNegAutoConfig | ATTRIBUTE | GET-SET | | | | | | | | | | | | | | | | | | | X |
| aAutoNegLocalTechnologyAbility | ATTRIBUTE | GET | | | | | | | | | | | | | | | | | | | X |
| aAutoNegAdvertisedTechnologyA- | ATTRIBUTE | GET-SET | | | | | | | | | | | | | | | | | | | X |
| aAutoNegReceivedTechnologyAbility | ATTRIBUTE | GET | | | | | | | | | | | | | | | | | | | X |
| aAutoNegLocalSelectorAbility | ATTRIBUTE | GET | | | | | | | | | | | | | | | | | | | X |
| aAutoNegAdvertisedSelectorAbility | ATTRIBUTE | GET-SET | | | | | | | | | | | | | | | | | | | X |
| aAutoNegReceivedSelectorAbility | ATTRIBUTE | GET | | | | | | | | | | | | | | | | | | | X |
| acAutoNegRestartAutoConfig | ACTION | | | | | | | | | | | | | | | | | | | | X |
| acAutoNegAdminControl | ACTION | | | | | | | | | | | | | | | | | | | | X |
| **Common Attributes Template** | | | | | | | | | | | | | | | | | | | | | |
| aCMCounter | ATTRIBUTE | GET | | X | X | X | | X | | X | | X | X | X | | X | X | | | X | |

## 30.3 Layer management for 10 Mb/s and 100 Mb/s DTEs

### 30.3.1 MAC entity managed object class

This subclause formally defines the behaviours for the oMACEntity managed object class attributes, actions, and notifications.

#### 30.3.1.1 MAC entity attributes

#### 30.3.1.1.1 aMACID

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
The value of aMACID is assigned so as to uniquely identify a MAC among the subordinate managed objects of the containing object.;

#### 30.3.1.1.2 aFramesTransmittedOK

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
A count of frames that are successfully transmitted. This counter is incremented when the TransmitStatus is reported as transmitOK. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

#### 30.3.1.1.3 aSingleCollisionFrames

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 13 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
A count of frames that are involved in a single collision, and are subsequently transmitted successfully. This counter is incremented when the result of a transmission is reported as transmitOK and the attempt value is 2. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

#### 30.3.1.1.4 aMultipleCollisionFrames

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 11 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
A count of frames that are involved in more than one collision and are subsequently transmitted successfully. This counter is incremented when the TransmitStatus is reported as transmitOK and the value of the attempts variable is greater than 2 and less or equal to attemptLimit. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

#### 30.3.1.1.5 aFramesReceivedOK

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
A count of frames that are successfully received (receiveOK). This does not include frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error. This counter is incremented when the ReceiveStatus is reported as receiveOK. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

#### 30.3.1.1.6 aFrameCheckSequenceErrors

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
A count of frames that are an integral number of octets in length and do not pass the FCS check. This counter is incremented when the ReceiveStatus is reported as frameCheckError. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

#### 30.3.1.1.7 aAlignmentErrors

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
A count of frames that are not an integral number of octets in length and do not pass the FCS check. This counter is incremented when the ReceiveStatus is reported as alignmentError. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

#### 30.3.1.1.8 aOctetsTransmittedOK

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 1 230 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
A count of data and padding octets of frames that are successfully transmitted. This counter is incremented when the TransmitStatus is reported as transmitOK. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

#### 30.3.1.1.9 aFramesWithDeferredXmissions

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 13 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames whose transmission was delayed on its first attempt because the medium was busy. This counter is incremented when the Boolean variable deferred has been asserted by the TransmitLinkMgmt function (4.2.8). Frames involved in any collisions are not counted. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

#### 30.3.1.1.10 aLateCollisions

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of the times that a collision has been detected later than 512 BT into the transmitted packet. A late collision is counted twice, i.e., both as a collision and as a lateCollision. This counter is incremented when the lateCollisionCount variable is nonzero. The actual update is incremented in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

#### 30.3.1.1.11 aFramesAbortedDueToXSColls

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 3255 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of the frames that, due to excessive collisions, are not transmitted successfully. This counter is incremented when the value of the attempts variable equals attemptLimit during a transmission. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

#### 30.3.1.1.12 aFramesLostDueToIntMACXmitError

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that would otherwise be transmitted by the station, but could not be sent due to an internal MAC sublayer transmit error. If this counter is incremented, then none of the other counters in this subclause is incremented. The exact meaning and mechanism for incrementing this counter is implementation dependent.;

#### 30.3.1.1.13 aCarrierSenseErrors

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of times that the carrierSense variable was not asserted or was deasserted during the transmission of a frame without collision (see 7.2.4.6). This counter is incremented when the carrierSenseFailure flag is true at the end of transmission. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

### 30.3.1.1.14 aOctetsReceivedOK

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 1 230 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
A count of data and padding octets in frames that are successfully received. This does not include octets in frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error. This counter is incremented when the result of a reception is reported as a receiveOK status. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

### 30.3.1.1.15 aFramesLostDueToIntMACRcvError

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
A count of frames that would otherwise be received by the station, but could not be accepted due to an internal MAC sublayer receive error. If this counter is incremented, then none of the other counters in this subclause is incremented. The exact meaning and mechanism for incrementing this counter is implementation dependent.;

### 30.3.1.1.16 aPromiscuousStatus

ATTRIBUTE

APPROPRIATE SYNTAX:
BOOLEAN

BEHAVIOUR DEFINED AS:
A GET operation returns the value "true" for promiscuous mode enabled, and "false" otherwise.

Frames without errors received solely because this attribute has the value "true" are counted as frames received correctly; frames received in this mode that do contain errors update the appropriate error counters.

A SET operation to the value "true" provides a means to cause the LayerMgmtRecognizeAddress function to accept frames regardless of their destination address.

A SET operation to the value "false" causes the MAC sublayer to return to the normal operation of carrying out address recognition procedures for station, broadcast, and multicast group addresses (LayerMgmtRecognizeAddress function).;

### 30.3.1.1.17 aReadMulticastAddressList

ATTRIBUTE

APPROPRIATE SYNTAX:
SEQUENCE OF MAC addresses

BEHAVIOUR DEFINED AS:
The current multicast address list.;

### 30.3.1.1.18 aMulticastFramesXmittedOK

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that are successfully transmitted, as indicated by the status value transmitOK, to a group destination address other than broadcast. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

### 30.3.1.1.19 aBroadcastFramesXmittedOK

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of the frames that were successfully transmitted as indicated by the TransmitStatus transmitOK, to the broadcast address. Frames transmitted to multicast addresses are not broadcast frames and are excluded. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

### 30.3.1.1.20 aFramesWithExcessiveDeferral

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 412 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that deferred for an excessive period of time. This counter may only be incremented once per LLC transmission. This counter is incremented when the excessDefer flag is set. The actual update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

### 30.3.1.1.21 aMulticastFramesReceivedOK

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that are successfully received and are directed to an active nonbroadcast group address. This does not include frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error. This counter is incremented as indicated by the receiveOK status, and the value in the destinationField. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

### 30.3.1.1.22 aBroadcastFramesReceivedOK

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames that are successfully received and are directed to the broadcast group address. This does not include frames received with frame-too-long, FCS, length or alignment errors, or frames lost due to internal MAC sublayer error. This counter is incremented as indicated by the receiveOK status, and the value in the destinationField. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

#### 30.3.1.1.23 aInRangeLengthErrors

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames with a length field value between the minimum unpadded LLC data size and the maximum allowed LLC data size, inclusive, that does not match the number of LLC data octets received. The counter also contains frames with a length field value less than the minimum unpadded LLC data size. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

#### 30.3.1.1.24 aOutOfRangeLengthField

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames with a length field value greater than the maximum allowed LLC data size. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

#### 30.3.1.1.25 aFrameTooLongErrors

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 815 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of frames received that exceed the maximum permitted frame size. This counter is incremented when the status of a frame reception is frameTooLong. The actual update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).;

#### 30.3.1.1.26 aMACEnableStatus

ATTRIBUTE

APPROPRIATE SYNTAX:

BOOLEAN

BEHAVIOUR DEFINED AS:

True if MAC sublayer is enabled and false if disabled. This is accomplished by setting or checking the values of the receiveEnabled and transmitEnabled variables.Setting to true provides a means to cause the MAC sublayer to enter the normal operational state at idle. The PLS is reset by this operation (see 7.2.2.2.1). This is accomplished by setting receiveEnabled and transmitEnabled to true.

Setting to false causes the MAC sublayer to end all transmit and receive operations, leaving it in a disabled state. This is accomplished by setting receiveEnabled and transmitEnabled to false.;

### 30.3.1.1.27 aTransmitEnableStatus

ATTRIBUTE

APPROPRIATE SYNTAX:
BOOLEAN

BEHAVIOUR DEFINED AS:
True if transmission is enabled and false otherwise. This is accomplished by setting or checking the value of the transmitEnabled variable.
Setting this to true provides a means to enable MAC sublayer frame transmission (TransmitFrame function). This is accomplished by setting transmitEnabled to true.
Setting this to false will inhibit the transmission of further frames by the MAC sublayer (TransmitFrame function). This is accomplished by setting transmitEnabled to false.;

### 30.3.1.1.28 aMulticastReceiveStatus

ATTRIBUTE

APPROPRIATE SYNTAX:
BOOLEAN

BEHAVIOUR DEFINED AS:
True if multicast receive is enabled, and false otherwise. Setting this to true provides a means to cause the MAC sublayer to return to the normal operation of multicast frame reception. Setting this to false will inhibit the reception of further multicast frames by the MAC sublayer.;

### 30.3.1.1.29 aReadWriteMACAddress

ATTRIBUTE

APPROPRIATE SYNTAX:
MACAddress

BEHAVIOUR DEFINED AS:
Read the MAC station address or change the MAC station address to the one supplied (RecognizeAddress function). Note that the supplied station address shall not have the group bit set and shall not be the null address.;

### 30.3.1.1.30 aCollisionFrames

ATTRIBUTE

APPROPRIATE SYNTAX:
A SEQUENCE of 32 generalized nonresettable counters. Each counter has a maximum increment rate of 13 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
A histogram of collision activity. The indices of this array (1 to attemptLimit – 1) denote the number of collisions experienced in transmitting a frame. Each element of this array contains a counter that denotes the number of frames that have experienced a specific number of collisions. When the TransmitStatus is reported as transmitOK and the value of the attempts variable equals n, then collisionFrames[n–1] counter is incremented. The elements of this array are incremented in the LayerMgmtTransmitCounters procedure (5.2.4.2).;

### 30.3.1.2 MAC entity actions

#### 30.3.1.2.1 acInitializeMAC

ACTION

APPROPRIATE SYNTAX:
None required

BEHAVIOUR DEFINED AS:
This action provides a means to call the Initialize procedure (4.2.7.5). This action also results in the initialization of the PLS.;

#### 30.3.1.2.2 acAddGroupAddress

ACTION

APPROPRIATE SYNTAX:
MACAddress

BEHAVIOUR DEFINED AS:
Add the supplied multicast group address to the address recognition filter (RecognizeAddress function).;

#### 30.3.1.2.3 acDeleteGroupAddress

ACTION

APPROPRIATE SYNTAX:
MACAddress

BEHAVIOUR DEFINED AS:
Delete the supplied multicast group address from the address recognition filter (RecognizeAddress function).;

#### 30.3.1.2.4 acExecuteSelfTest

ACTION

APPROPRIATE SYNTAX:
None required

BEHAVIOUR DEFINED AS:
Execute a self-test and report the results (success or failure). The actual mechanism employed to carry out the self-test is not defined in this standard. If a clause 22 MII is present then this action shall also invoke a data integrity test using MII loopback, returning to normal operation on completion of the test.;

### 30.3.2 PHY entity managed object class

This subclause formally defines the behaviours for the oPHYEntity managed object class attributes, actions and notifications. Management of that portion of the physical sublayer whose physical containment within the DTE is optional is outside the scope of this clause.

### 30.3.2.1 PHY entity attributes

#### 30.3.2.1.1 aPHYID

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
The value of aPHYID is assigned so as to uniquely identify a PHY, i.e., Physical Layer among the subordinate managed objects of system (systemID and system are defined in ISO/IEC 10165-2: 1992 [SMI], Definition of management information).;

#### 30.3.2.1.2 aPhyType

ATTRIBUTE

APPROPRIATE SYNTAX:
An ENUMERATED VALUE that has one of the following entries:

| | |
|---|---|
| other | Undefined |
| unknown | Initializing, true state or type not yet known |
| none | MII present and nothing connected |
| 10 Mb/s | Clause 7 10 Mb/s Manchester |
| 100BASE-T4 | Clause 23 100 Mb/s 8B/6T |
| 100BASE-X | Clause 24 100 Mb/s 4B/5B |

BEHAVIOUR DEFINED AS:
A read-only value that identifies the PHY type. The enumeration of the type is such that the value matches the clause number of the standard that specifies the particular PHY. The value of this attribute maps to the value of aMAUType. The enumeration "none" can only occur in a standard implementation an MII exists and there is nothing connected. However, the attribute aMIIDetect should be used to determine whether an MII exists or not.;

#### 30.3.2.1.3 aPhyTypeList

ATTRIBUTE

APPROPRIATE SYNTAX:
A SEQUENCE that meets the requirements of the description below:

| | |
|---|---|
| other | Undefined |
| unknown | Initializing, true state or type not yet known |
| none | MII present and nothing connected |
| 10 Mb/s | Clause 7 10 Mb/s Manchester |
| 100BASE-T4 | Clause 23 100 Mb/s 8B/6T |
| 100BASE-X | Clause 24 100 Mb/s 4B/5B |

BEHAVIOUR DEFINED AS:
A read-only list of the possible types that the PHY could be, identifying the ability of the PHY. If clause 28, Auto-Negotiation, is present, then this attribute will map to the local technology ability.;

#### 30.3.2.1.4 aSQETestErrors

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 16 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A count of times that the SQE_TEST_ERROR was received. The SQE_TEST_ERROR is set in accordance with the rules for verification of the SQE detection mechanism in the PLS Carrier Sense function (see 7.2.4.6). The SQE test function is not a part of 100 Mb/s PHY operation, and so SQETestErrors will not occur in 100 Mb/s PHYs.;

#### 30.3.2.1.5 aSymbolErrorDuringCarrier

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 160 000 counts per second for 100 Mb/s implementations

BEHAVIOUR DEFINED AS:

A count of the number of times when valid carrier was present and there was at least one occurrence of an invalid data symbol. This can increment only once per valid carrier event. If a collision is present this attribute will not increment.;

#### 30.3.2.1.6 aMIIDetect

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has one of the following entries:
unknown
present, nothing connected
present, connected
absent

BEHAVIOUR DEFINED AS:

An attribute of the PhyEntity managed object class indicating whether an MII connector is physically present, and if so whether it is detectably connected as specified in 22.2.2.12.;

#### 30.3.2.1.7 aPhyAdminState

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE that has the following entries:
disabled
enabled

BEHAVIOUR DEFINED AS:

A disabled PHY neither transmits nor receives. The PHY shall be explicitly enabled to restore operation. The acPhyAdminControl action provides this ability. The port enable/disable function as reported by this attribute is preserved across DTE reset including loss of power. Only one PHY per MAC can be enabled at any one time. Setting a PHY to the enabled state using the action acPhyAdminControl will result in all other instances of PHY (indicated by PhyID) instantiated within the same MAC to be disabled. If a clause 22 MII is present then setting this attribute to "disable" will result in electrical isolation as defined in 22.2.4.1.6, Isolate; and setting this attribute to "enabled" will result in normal operation as defined in 22.2.4.1.5, Power down; and 22.2.4.1.6, Isolate.;

#### 30.3.2.2 PHY entity actions

#### 30.3.2.2.1 acPhyAdminControl

ACTION

APPROPRIATE SYNTAX:
Same as aPortAdminState

BEHAVIOUR DEFINED AS:
This action provides a means to alter aPhyAdminState. Setting a PHY to the enabled state will result in all other instances of PHY being disabled.;

## 30.4 Layer management for 10 Mb/s and 100 Mb/s baseband repeaters

### 30.4.1 Repeater managed object class

This subclause formally defines the behaviours for the oRepeater managed object class, attributes, actions, and notifications.

#### 30.4.1.1 Repeater attributes

#### 30.4.1.1.1 aRepeaterID

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
The value of aRepeaterID is assigned so as to uniquely identify a repeater among the subordinate managed objects of system (systemID and system are defined in ISO/IEC 10165-2: 1992 [SMI], Definition of management information).;

#### 30.4.1.1.2 aRepeaterType

ATTRIBUTE

APPROPRIATE SYNTAX:
An INTEGER that meets the requirements of the description below:

| | |
|---|---|
| 9 | 10 Mb/s Baseband |
| 271 | 100 Mb/s Baseband, Class I |
| 272 | 100 Mb/s Baseband, Class II |
| other | See 20.2.2.3 |
| unknown | Initializing, true state or type not yet known |

BEHAVIOUR DEFINED AS:
Returns a value that identifies the CSMA/CD repeater type. The enumeration of the type is such that the value matches the clause number of the standard that specifies the particular repeater, with further numerical identification for the repeater classes within the same clause.;

### 30.4.1.1.3 aRepeaterGroupCapacity

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
The aRepeaterGroupCapacity is the number of groups that can be contained within the repeater. Within each managed repeater, the groups are uniquely numbered in the range from 1 to aRepeaterGroupCapacity.

Some groups may not be present in a given repeater instance, in which case the actual number of groups present is less than aRepeaterGroupCapacity. The number of groups present is never greater than aRepeaterGroupCapacity.;

### 30.4.1.1.4 aGroupMap

ATTRIBUTE

APPROPRIATE SYNTAX:
BITSTRING

BEHAVIOUR DEFINED AS:
A string of bits which reflects the current configuration of units that are viewed by group managed objects. The length of the bitstring is "aRepeaterGroupCapacity" bits. The first bit relates to group 1. A "1" in the bitstring indicates presence of the group, "0" represents absence of the group.;

### 30.4.1.1.5 aRepeaterHealthState

ATTRIBUTE

APPROPRIATE SYNTAX:
An ENUMERATED VALUE LIST that has the following entries:

| | |
|---|---|
| other | undefined or unknown |
| ok | no known failures |
| repeaterFailure | known to have a repeater related failure |
| groupFailure | known to have a group related failure |
| portFailure | known to have a port related failure |
| generalFailure | has a failure condition, unspecified type |

BEHAVIOUR DEFINED AS:
The aRepeaterHealthState attribute indicates the operational state of the repeater. The aRepeaterHealthData and aRepeaterHealthText attributes may be consulted for more specific information about the state of the repeater's health. In case of multiple kinds of failures (e.g., repeater failure and port failure), the value of this attribute shall reflect the highest priority in the following order:
repeater failure
group failure
port failure
general failure;

### 30.4.1.1.6 aRepeaterHealthText

ATTRIBUTE

APPROPRIATE SYNTAX:
A PrintableString, 255 characters max

BEHAVIOUR DEFINED AS:
The aRepeaterHealthText attribute is a text string that provides information relevant to the operational state of the repeater. Repeater vendors may use this mechanism to provide detailed failure information or instructions for problem resolution.

The contents are vendor specific.;

### 30.4.1.1.7 aRepeaterHealthData

ATTRIBUTE

APPROPRIATE SYNTAX:
OCTET STRING, 0–255

BEHAVIOUR DEFINED AS:
The aRepeaterHealthData attribute is a block of data octets that provides information relevant to the operational state of the repeater. The encoding of this data block is vendor dependent. Repeater vendors may use this mechanism to provide detailed failure information or instructions for problem resolution.;

### 30.4.1.1.8 aTransmitCollisions

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
For a clause 9 repeater, the counter increments every time the repeater state diagram enters the TRANSMIT COLLISION state from any state other than ONE PORT LEFT (figure 9-2). For a clause 27 repeater, the counter increments every time the Repeater Core state diagram enters the JAM state as a result of Activity(ALL) > 1 (figure 27-2).;

### 30.4.1.2 Repeater actions

### 30.4.1.2.1 acResetRepeater

ACTION

APPROPRIATE SYNTAX:
None required

BEHAVIOUR DEFINED AS:
This causes a transition to the START state of figure 9-2 for a clause 9 repeater, or to the START state of figure 27-2 for a clause 27 repeater. The repeater performs a disruptive self-test that has the following characteristics:
1. The components are not specified
2. The test resets the repeater but without affecting management information about the repeater
3. The test does not inject packets onto any segment
4. Packets received during the test may or may not be transferred
5. The test does not interfere with management functions

This causes a nRepeaterReset notification to be sent.;

#### 30.4.1.2.2 acExecuteNonDisruptiveSelfTest

ACTION

APPROPRIATE SYNTAX:
None required

BEHAVIOUR DEFINED AS:
The repeater performs a vendor-specific, non-disruptive self-test that has the following characteristics:
1. The components are not specified
2. The test does not change the state of the repeater or management information about the repeater
3. The test does not inject packets onto any segment
4. The test does not prevent the transfer of any packets
5. Completion of the test causes a nRepeaterHealth to be sent.;

### 30.4.1.3 Repeater notifications

#### 30.4.1.3.1 nRepeaterHealth

NOTIFICATION

APPROPRIATE SYNTAX:
A SEQUENCE of three data types. The first is mandatory, the following two are optional. The first is the value of the attribute aRepeaterHealthState. The second is the value of the attribute aRepeaterHealthText. The third is the value of the attribute aRepeaterHealthData

BEHAVIOUR DEFINED AS:
This notification conveys information related to the operational state of the repeater. See the aRepeaterHealthState, aRepeaterHealthText, and aRepeaterHealthData attributes for descriptions of the information that is sent.

The nRepeaterHealth notification is sent only when the health state of the repeater changes. The nRepeaterHealth notification shall contain repeaterHealthState. repeaterHealthData and repeaterHealthText may or may not be included. The nRepeaterHealth notification is not sent as a result of powering up a repeater.;

#### 30.4.1.3.2 nRepeaterReset

NOTIFICATION

APPROPRIATE SYNTAX:
A SEQUENCE of three data types. The first is mandatory, the following two are optional. The first is the value of the attribute aRepeaterHealthState. The second is the value of the attribute aRepeaterHealthText. The third is the value of the attribute aRepeaterHealthData

BEHAVIOUR DEFINED AS:
This notification conveys information related to the operational state of the repeater. The nRepeaterReset notification is sent when the repeater is reset as the result of a power-on condition or upon completion of the acResetRepeater action. The nRepeaterReset notification shall contain repeaterHealthState. repeaterHealthData and repeaterHealthText may or may not be included.;

#### 30.4.1.3.3 nGroupMapChange

NOTIFICATION

APPROPRIATE SYNTAX:
BITSTRING

BEHAVIOUR DEFINED AS:
This notification is sent when a change occurs in the group structure of a repeater. This occurs only when a group is logically removed from or added to a repeater. The nGroupMapChange notification is not sent when powering up a repeater. The value of the notification is the updated value of the aGroupMap attribute.;

### 30.4.2 Group managed object class

This subclause formally defines the behaviours for the oGroup managed object class, attributes, actions, and notifications.

#### 30.4.2.1 Group attributes

##### 30.4.2.1.1 aGroupID

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
A value unique within the repeater. The value of aGroupID is assigned so as to uniquely identify a group among the subordinate managed objects of the containing object (oRepeater). This value is never greater than aRepeaterGroupCapacity.;

##### 30.4.2.1.2 aGroupPortCapacity

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
The aGroupPortCapacity is the number of ports contained within the group. Valid range is 1–1024. Within each group, the ports are uniquely numbered in the range from 1 to aGroupPortCapacity. Some ports may not be present in a given group instance, in which case the actual number of ports present is less than aGroupPortCapacity. The number of ports present is never greater than aGroupPortCapacity.;

##### 30.4.2.1.3 aPortMap

ATTRIBUTE

APPROPRIATE SYNTAX:
BitString

BEHAVIOUR DEFINED AS:
A string of bits that reflects the current configuration of port managed objects within this group. The length of the bitstring is “aGroupPortCapacity” bits. The first bit relates to group 1. A “1” in the bitstring indicates presence of the port, “0” represents absence of the port.;

#### 30.4.2.2 Group notifications

##### 30.4.2.2.1 nPortMapChange

NOTIFICATION

APPROPRIATE SYNTAX:
BitString

BEHAVIOUR DEFINED AS:

This notification is sent when a change occurs in the port structure of a group. This occurs only when a port is logically removed from or added to a group. The nPortMapChange notification is not sent when powering up a repeater. The value of the notification is the updated value of the aPortMap attribute.;

### 30.4.3 Repeater port managed object class

This subclause formally defines the behaviours for the oRepeaterPort managed object class, attributes, actions, and notifications.

#### 30.4.3.1 Port attributes

##### 30.4.3.1.1 aPortID

ATTRIBUTE

APPROPRIATE SYNTAX:

INTEGER

BEHAVIOUR DEFINED AS:

A value unique in the group. It is assumed that ports are partitioned into groups that also have IDs. The value of aPortID is assigned so as to uniquely identify a repeater port among the subordinate managed objects of the containing object (oGroup). This value can never be greater than aGroupPortCapacity.;

##### 30.4.3.1.2 aPortAdminState

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE LIST that has the following entries:
disabled
enabled

BEHAVIOUR DEFINED AS:

A disabled port neither transmits nor receives. The port shall be explicitly enabled to restore operation. The acPortAdminControl action provides this ability. The port enable/disable function as reported by this attribute is preserved across repeater reset including loss of power. aPortAdminState takes precedence over auto-partition and functionally operates between the auto-partition mechanism and the AUI/PMA. Auto-partition is reinitialized whenever acPortAdminControl is enabled.;

##### 30.4.3.1.3 aAutoPartitionState

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED VALUE LIST that has the following entries:
autoPartitioned
notAutoPartitioned

BEHAVIOUR DEFINED AS:

The aAutoPartitionState flag indicates whether the port is currently partitioned by the repeater's auto-partition protection. The conditions that cause port partitioning are specified in partition state diagram in clauses 9 and 27. They are not differentiated here. A clause 27 repeater port partitions on entry to the PARTITION WAIT state of the partition state diagram (figure 27-8).;

##### 30.4.3.1.4 aReadableFrames

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 15 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

A representation of the total frames of valid frame length. Increment counter by one for each frame whose OctetCount is greater than or equal to minFrameSize and less than or equal to maxFrameSize (see 4.4.2.1) and for which the FCSError and CollisionEvent signals are not asserted.

NOTE—This statistic provides one of the parameters necessary for obtaining the packet error rate.;

#### 30.4.3.1.5 aReadableOctets

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 1 240 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by OctetCount for each frame which has been determined to be a readable frame.

NOTE—This statistic provides an indicator of the total data transferred.;

#### 30.4.3.1.6 aFrameCheckSequenceErrors

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 15 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by one for each frame with the FCSError signal asserted and the FramingError and CollisionEvent signals deasserted and whose OctetCount is greater than or equal to minFrameSize and less than or equal to maxFrameSize (see 4.4.2.1).;

#### 30.4.3.1.7 aAlignmentErrors

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 15 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by one for each frame with the FCSError and FramingError signals asserted and CollisionEvent signal deasserted and whose OctetCount is greater than or equal to minFrameSize and less than or equal to maxFrameSize (see 4.4.2.1). If aAlignmentErrors is incremented then the aFrameCheckSequenceErrors attribute shall not be incremented for the same frame.;

#### 30.4.3.1.8 aFramesTooLong

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 815 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by one for each frame whose OctetCount is greater than maxFrameSize (see 4.4.2.1). If aFrameTooLong is counted then neither the aAlignmentErrors nor the aFrameCheckSequenceErrors attribute shall be incremented for the frame.;

#### 30.4.3.1.9 aShortEvents

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by one for each CarrierEvent with ActivityDuration less than ShortEventMaxTime. In the 10 Mb/s case ShortEventMaxTime is greater than 74 BT and less than 82 BT. ShortEventMaxTime has tolerances included to provide for circuit losses between a conformance test point at the AUI and the measurement point within the state diagram. In the 100 Mb/s case ShortEventMaxTime is 84 bits (21 nibbles).

NOTES

1—shortEvents may indicate externally generated noise hits that will cause the repeater to transmit Runts to its other ports, or propagate a collision (which may be late) back to the transmitting DTE and damaged frames to the rest of the network.

2—Implementors may wish to consider selecting the ShortEventMaxTime towards the lower end of the allowed tolerance range to accommodate bit losses suffered through physical channel devices not budgeted for within this standard.

3—Note also that the significance of this attribute is different in 10 and 100 Mb/s collision domains. Clause 9 repeaters perform fragment extension of short events that would be counted as runts on the interconnect ports of other repeaters. Clause 27 repeaters do not perform fragment extension.;

#### 30.4.3.1.10 aRunts

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by one for each CarrierEvent that meets one of the following two conditions. Only one test need be made. a) The ActivityDuration is greater than ShortEventMaxTime and less than ValidPacketMinTime and the CollisionEvent signal is de-asserted (10 Mb/s operation) or the COLLISION COUNT INCREMENT state of the partition state diagram (figure 27-8) has not been entered (100 Mb/s operation). b) The OctetCount is less than 64, the ActivityDuration is greater than ShortEventMaxTime, and the CollisionEvent signal is de-asserted (10 Mb/s operation), or the COLLISION COUNT INCREMENT state of the partition state diagram (figure 27-8) has not been entered (100 Mb/s operation). ValidPacketMinTime is greater than or equal to 552 BT and less than 565 BT. At 10 Mb/s an event whose length is greater than 74 BT but less than 82 BT shall increment either the aShortEvents attribute or the aRunts attribute, but not both. A CarrierEvent greater than or equal to 552 BT but less than 565 BT may or may not be counted as a runt. ValidPacketMinTime has tolerances included to provide for circuit losses between a conformance test point at the AUI and the measurement point within the state diagram.

NOTE—Runts usually indicate collision fragments, a normal network event.
In certain situations associated with large diameter networks a percentage of runts may exceed ValidPacketMinTime.;

### 30.4.3.1.11 aCollisions

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

For a clause 9 repeater the counter increments for any CarrierEvent signal on any port in which the CollisionEvent signal on this port is asserted. For a clause 27 repeater port the counter increments on entering the COLLISION COUNT INCREMENT state of the partition state diagram (figure 27-8).;

### 30.4.3.1.12 aLateEvents

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 75 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

For a clause 9 repeater port this counter increments for each CarrierEvent in which the CollIn(X) variable transitions to the value SQE (see 9.6.6.2) while the ActivityDuration is greater than the LateEventThreshold. For a clause 27 repeater port this counter increments on entering the COLLISION COUNT INCREMENT state of the partition state diagram (figure 27-8) while the ActivityDuration is greater than the LateEventThreshold. Such a CarrierEvent is counted twice, as both a aCollision and as a aLateEvent.

The LateEventThreshold is greater than 480 BT and less than 565 BT. LateEventThreshold has tolerances included to permit an implementation to build a single threshold to serve as both the LateEventThreshold and ValidPacketMinTime threshold.;

### 30.4.3.1.13 aVeryLongEvents

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 250 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:

For a clause 9 repeater port this counter increments for each CarrierEvent whose ActivityDuration is greater than the MAU Jabber Lockup Protection timer TW3 (see 9.6.1, 9.6.5). For a clause 27 repeater port this counter increments on entry to the RX JABBER state of the receive timer state diagram (figure 27-7). Other counters may be incremented as appropriate.;

### 30.4.3.1.14 aDataRateMismatches

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter

BEHAVIOUR DEFINED AS:

Increment counter by one for each frame received by this port that meets all of the conditions required by only one of the following two measurement methods: Measurement method A: 1) The CollisionEvent signal is not asserted (10 Mb/s operation) or the COLLISION COUNT INCREMENT state of the partition state diagram (figure 27-8) has not been entered (100 Mb/s operation). 2) The ActivityDuration is greater than ValidPacketMinTime. 3) The frequency (data rate) is detectably mismatched from the local transmit frequency. Measurement method B: 1) The CollisionEvent signal is not asserted (10 Mb/s operation) or the COLLISION COUNT INCREMENT state of the partition state diagram (figure 27-8) has not been entered (100 Mb/s operation). 2) The OctetCount is greater than 63. 3) The frequency (data rate) is detectably mismatched from the local transmit frequency. The exact degree of mismatch is vendor specific and is to be defined by the vendor for conformance testing.

When this event occurs, other counters whose increment conditions were satisfied may or may not also be incremented, at the implementor's discretion.

NOTE—Whether or not the repeater was able to maintain data integrity is beyond the scope of this standard.;

### 30.4.3.1.15 aAutoPartitions

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter

BEHAVIOUR DEFINED AS:

Increment counter by one for each time that the repeater has automatically partitioned this port. The conditions that cause a clause 9 repeater port to partition are specified in the partition state diagram in clause 9. They are not differentiated here. A clause 27 repeater port partitions on entry to the PARTITION WAIT state of the partition state diagram (figure 27-8).;

### 30.4.3.1.16 aIsolates

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 400 counts per second at 100 Mb/s

BEHAVIOUR DEFINED AS:

Increment counter by one each time that the repeater port automatically isolates as a consequence of false carrier events. The conditions that cause a port to automatically isolate are as defined by the transition from the FALSE CARRIER state to the LINK UNSTABLE state of the carrier integrity state diagram (figure 27-9).

NOTE—Isolates do not affect the value of aPortAdminState.;

### 30.4.3.1.17 aSymbolErrorDuringPacket

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 160 000 counts per second for 100 Mb/s implementations

BEHAVIOUR DEFINED AS:

A count of the number of times when valid length packet was received at the port and there was at least one occurrence of an invalid data symbol. This can increment only once per valid carrier event. A collision presence at any port of the repeater containing port N, will not cause this attribute to increment.;

#### 30.4.3.1.18 aLastSourceAddress

ATTRIBUTE

APPROPRIATE SYNTAX:
MACAddress

BEHAVIOUR DEFINED AS:
The Source Address of the last readable Frame received by this port.;

#### 30.4.3.1.19 aSourceAddressChanges

ATTRIBUTE

APPROPRIATE SYNTAX:
Generalized nonresettable counter. This counter has a maximum increment rate of 15 000 counts per second at 10 Mb/s

BEHAVIOUR DEFINED AS:
Increment counter by one each time that the aLastSourceAddress attribute has changed.

NOTE—This may indicate whether a link is connected to a single DTE or another multi-user segment.;

### 30.4.3.2 Port actions

#### 30.4.3.2.1 acPortAdminControl

ACTION

APPROPRIATE SYNTAX:
Same as aPortAdminState

BEHAVIOUR DEFINED AS:
This action provides a means to alter aPortAdminState and exert a BEGIN on the Partitioning state diagram (figure 9-6) or the Partition state diagram (figure 27-8) upon taking the value "enabled".;

## 30.5 Layer management for 10 Mb/s and 100 Mb/s MAUs

### 30.5.1 MAU managed object class

This subclause formally defines the behaviours for the oMAU managed object class, attributes, actions, and notifications.

#### 30.5.1.1 MAU attributes

#### 30.5.1.1.1 aMAUID

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
The value of aMAUID is assigned so as to uniquely identify a MAU among the subordinate managed objects of the containing object.;

### 30.5.1.1.2 aMAUType

ATTRIBUTE

APPROPRIATE SYNTAX:

A GET-SET ENUMERATION that meets the requirements of the description below:

| | |
|---|---|
| global | Reserved for future use |
| other | See 20.2.2.3 |
| unknown | Initializing, true state or type not yet known |
| AUI | no internal MAU, view from AUI |
| 10BASE5 | Thick coax MAU as specified in clause 8 |
| FOIRL | FOIRL MAU as specified in 9.9 |
| 10BASE2 | Thin coax MAU as specified in clause 10 |
| 10BROAD36 | Broadband DTE MAU as specified in clause 11 |
| 10BASE-T | UTP MAU as specified in clause 14 |
| 10BASE-FP | Passive fiber MAU as specified in clause 16 |
| 10BASE-FB | Synchronous fiber MAU as specified in clause 17 |
| 10BASE-FL | Asynchronous fiber MAU as specified in clause 18 |
| 100BASE-T4 | Four-pair Category 3 UTP as specified in clause 23 |
| 100BASE-TX | Two-pair Category 5 UTP as specified in clause 25 |
| 100BASE-FX | X fiber over PMD as specified in clause 26 |
| 802.9a | Integrated services MAU as specified in IEEE Std 802.9 ISLAN-16T |

BEHAVIOUR DEFINED AS:

Returns a value that identifies the 10 Mb/s or 100 Mb/s internal MAU type. The enumeration of the type is such that the value matches the clause number of the standard that specifies the particular MAU. If an AUI is to be identified to access an external MAU, then type "AUI" is returned. A SET operation to one of the possible enumerations indicated by aMAUTypeList will force the MAU into the new operating mode. If a clause 22 MII is present, then this will map to the mode force bits specified in 22.2.4.1. If clause 28, Auto-Negotiation, is operational, then this will change the advertised ability to the single enumeration specified in the SET operation, and cause an immediate link renegotiation. A change in MAU type will also be reflected in oPHYType.;

### 30.5.1.1.3 aMAUTypeList

ATTRIBUTE

APPROPRIATE SYNTAX:

A SEQUENCE of ENUMERATIONS that match the syntax of aMAUType

BEHAVIOUR DEFINED AS:

A GET attribute that returns the possible types that the MAU could be, identifying the ability of the MAU. If clause 28 Auto-Negotiation is present, then this attribute will map to the local technology ability. This attribute maps to aPHYTypeList.;

### 30.5.1.1.4 aMediaAvailable

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED value list that has the following entries:

| | |
|---|---|
| other | undefined |
| unknown | initializing, true state not yet known |
| available | link or light normal, loopback normal |
| not available | link loss or low light, no loopback |
| remote fault | remote fault with no detail |
| invalid signal | invalid signal, applies only to 10BASE-FB |
| remote jabber | remote fault, reason known to be jabber |
| remote link loss | remote fault, reason known to be far-end link loss |
| remote test | remote fault, reason known to be test |

BEHAVIOUR DEFINED AS:

If the MAU is a link or fiber type (FOIRL, 10BASE-T, 10BASE-F), then this is equivalent to the link test fail state/low light function. For an AUI , 10BASE2, 10BASE5, or 10BROAD36 MAU, this indicates whether or not loopback is detected on the DI circuit. The value of this attribute persists between packets for MAU types AUI, 10BASE5, 10BASE2, 10BROAD36, and 10BASE-FP.

At power-up or following a reset, the value of this attribute will be "unknown" for AUI, 10BASE5, 10BASE2, 10BROAD36, and 10BASE-FP MAUs. For these MAUs loopback will be tested on each transmission during which no collision is detected. If DI is receiving *input* when DO returns to IDL after a transmission and there has been no collision during the transmission, then loopback will be detected. The value of this attribute will only change during noncollided transmissions for AUI, 10BASE2, 10BASE5, 10BROAD36, and 10BASE-FP MAUs.

For 100BASE-T4, 100BASE-TX, and 100BASE-FX the enumerations match the states within the respective link integrity state diagrams, figure 23-12 and 24-15. Any MAU that implements management of clause 28 Auto-Negotiation will map remote fault indication to MediaAvailable remote fault.

The enumeration "remote fault" applies to 10BASE-FB, 100BASE-X, far-end fault indication and non-specified remote faults from a system running clause 28 Auto-Negotiation. The enumerations "remote jabber," "remote link loss," or "remote test" should be used instead of "remote fault" where the reason for remote fault is identified in the remote signaling protocol.

Where a clause 22 MII is present, a logic one in the remote fault bit (22.2.4.2.9) maps to the enumeration "remote fault," a logic zero in the link status bit (22.2.4.2.11) maps to the enumeration "not available." The enumeration "not available" takes precedence over "remote fault.";

### 30.5.1.1.5 aLoseMediaCounter

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 10 counts per second

BEHAVIOUR DEFINED AS:

Counts the number of times that the MAU leaves
MediaAvailState "available." Mandatory for MAU type "AUI," optional for all others.;

### 30.5.1.1.6 aJabber

ATTRIBUTE

APPROPRIATE SYNTAX:

A SEQUENCE of two indications. The first, JabberFlag, consists of an ENUMERATED value list that has the following entries:

| | |
|---|---|
| other | undefined |
| unknown | initializing, true state not yet known |
| normal | state is true or normal |
| fault | state is false, fault, or abnormal |

The second, jabberCounter, is a generalized nonresettable counter. This counter has a maximum increment rate of 40 counts per second

BEHAVIOUR DEFINED AS:

If the MAU is in the JABBER state, the jabberFlag portion of the attribute is set to the "fault" value. The jabberCounter portion of the attribute is incremented each time the flag is set to the "fault" value. This attribute returns the value "other" for type AUI. Note that this counter will not increment for a 100 Mb/s PHY, as there is no defined JABBER state.;

### 30.5.1.1.7 aMAUAdminState

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED value list that has the following entries:

| | |
|---|---|
| other | undefined |
| unknown | initializing, true state not yet known |
| operational | powered and connected |
| standby | inactive but on |
| shutdown | similar to power down |

BEHAVIOUR DEFINED AS:

A MAU in management state “standby” forces DI and CI to idle and the media transmitter to idle or fault, if supported. The management state “standby” only applies to link type MAUs. The state of MediaAvailable is unaffected. A MAU or AUI in the management state “shutdown” assumes the same condition on DI, CI and the media transmitter as if it were powered down or not connected. For an AUI, this management state will remove power from the AUI. The MAU may return the value “undefined” for Jabber and MediaAvailable attributes when it is in this management state. A MAU in the management state “operational” is fully functional, and operates and passes signals to its attached DTE or repeater port in accordance with its specification.;

### 30.5.1.1.8 aBbMAUXmitRcvSplitType

ATTRIBUTE

APPROPRIATE SYNTAX:

An ENUMERATED value list that has the following entries:

| | |
|---|---|
| other | undefined |
| single | single-cable system |
| dual | dual-cable system, offset normally zero |

BEHAVIOUR DEFINED AS:

Returns a value that indicates the type of frequency multiplexing/cabling system used to separate the transmit and receive paths for the 10BROAD36 MAU. All other types return “undefined.”;

### 30.5.1.1.9 aBroadbandFrequencies

ATTRIBUTE

APPROPRIATE SYNTAX:

A SEQUENCE of two instances of the type INTEGER.

The first INTEGER represents the Transmitter Carrier Frequency. The value of its INTEGER represents the frequency of the carrier divided by 250 kHz.

The second INTEGER represents the Translation Offset Frequency. The value of its INTEGER represents the frequency of the offset divided by 250 kHz

BEHAVIOUR DEFINED AS:

Returns a value that indicates the transmit carrier frequency and translation offset frequency in MHz/4 for the 10BROAD36 MAU. This allows the frequencies to be defined to a resolution of 250 kHz.;

### 30.5.1.1.10 aFalseCarriers

ATTRIBUTE

APPROPRIATE SYNTAX:

Generalized nonresettable counter. This counter has a maximum increment rate of 160 000 counts per second under maximum network load, and 10 counts per second under zero network load, for 100 Mb/s implementations

BEHAVIOUR DEFINED AS:

A count of the number of false carrier events during IDLE in 100BASE-X links. This counter does not increment at the symbol rate. It can increment after a valid carrier completion at a maximum rate of once per 100 ms until the next carrier event.;

#### 30.5.1.2 MAU actions

#### 30.5.1.2.1 acResetMAU

ACTION

APPROPRIATE SYNTAX:

None required

BEHAVIOUR DEFINED AS:

Resets the MAU in the same manner as would a power-off, power-on cycle of at least 0.5 s duration. During the 0.5 s DO, DI, and CI should be idle.;

#### 30.5.1.2.2 acMAUAdminControl

ACTION

APPROPRIATE SYNTAX:

The same as used for aMAUAdminState

BEHAVIOUR DEFINED AS:

Executing an acMAUAdminControl action causes the MAU to assume the aMAUAdminState attribute value of one of the defined valid management states for control input. The valid inputs are "standby," "operational," and "shutdown" state (see the behaviour definition bMAUAdminState for the description of each of these states) except that a "standby" action to a mixing type MAU or an AUI will cause the MAU to enter the "shutdown" management state.;

#### 30.5.1.3 MAU notifications

#### 30.5.1.3.1 nJabber

NOTIFICATION

APPROPRIATE SYNTAX:

The same as used for aJabber

BEHAVIOUR DEFINED AS:

The notification is sent whenever a managed MAU enters the JABBER state.;

## 30.6 Management for link Auto-Negotiation

### 30.6.1 Auto-Negotiation managed object class

This subclause formally defines the behaviours for the oAuto-Negotiation managed object class, attributes, actions, and notifications.

### 30.6.1.1 Auto-Negotiation attributes

#### 30.6.1.1.1 aAutoNegID

ATTRIBUTE

APPROPRIATE SYNTAX:
INTEGER

BEHAVIOUR DEFINED AS:
The value of aAutoNegID is assigned so as to uniquely identify an Auto-Negotiation managed object among the subordinate managed objects of the containing object.;

#### 30.6.1.1.2 aAutoNegAdminState

ATTRIBUTE

APPROPRIATE SYNTAX:
An ENUMERATED VALUE that has one of the following entries:
enabled
disabled

BEHAVIOUR DEFINED AS:
An interface which has Auto-Negotiation signaling ability will be enabled to do so when this attribute is in the enabled state. If disabled then the interface will act as it would if it had no Auto-Negotiation signaling. Under these conditions it will immediately be forced to the states indicated by a write to the attribute aMAUType.;

#### 30.6.1.1.3 aAutoNegRemoteSignaling

ATTRIBUTE

APPROPRIATE SYNTAX:
An ENUMERATED VALUE that has one of the following entries:
detected
notdetected

BEHAVIOUR DEFINED AS:
The value indicates whether the remote end of the link is operating Auto-Negotiation signaling or not. It shall take the value detected if, during the previous link negotiation, FLP Bursts were received from the remote end.;

#### 30.6.1.1.4 aAutoNegAutoConfig

ATTRIBUTE

APPROPRIATE SYNTAX:
An ENUMERATED VALUE that has one of the following entries:
other
configuring
complete
disabled
parallel detect fail

BEHAVIOUR DEFINED AS:
Indicates whether Auto-Negotiation signaling is in progress or has completed. The enumeration "parallel detect fail" maps to a failure in parallel detection as defined in 28.2.3.1.;

### 30.6.1.1.5 aAutoNegLocalTechnologyAbility

ATTRIBUTE

APPROPRIATE SYNTAX:

A SEQUENCE that meets the requirements of the description below:

| | |
|---|---|
| global | Reserved for future use |
| other | Undefined |
| unknown | Initializing, true ability not yet known |
| 10BASE-T | 10BASE-T as defined in clause 14 |
| 10BASE-TFD | Full-duplex 10BASE-T |
| 100BASE-TX | 100BASE-TX as defined in clause 25 |
| 100BASE-TXFD | Full-duplex 100BASE-TX |
| 100BASE-T4 | 100BASE-T4 as defined in clause 23 |
| isoethernet | IEEE Std 802.9 ISLAN-16T |

BEHAVIOUR DEFINED AS:

This indicates the technology ability of the local hardware, as defined in clause 28.;

### 30.6.1.1.6 aAutoNegAdvertisedTechnologyAbility

ATTRIBUTE

APPROPRIATE SYNTAX:

Same as aAutoNegLocalTechnologyAbility

BEHAVIOUR DEFINED AS:

This GET-SET attribute maps to the Technology Ability Field of the Auto-Negotiation Link Code Word, defined in clause 28. A SET operation to a value not available in aAutoNegLocalTechnologyAbility will be rejected. A successful set operation will result in immediate link renegotiation if aAutoNegAdminState is enabled.

NOTE—This will in every case cause temporary link loss during link renegotiation. If set to a value incompatible with aAutoNegReceivedTechnologyAbility, link negotiation will not be successful and will cause permanent link loss.;

### 30.6.1.1.7 aAutoNegReceivedTechnologyAbility

ATTRIBUTE

APPROPRIATE SYNTAX:

Same as aAutoNegLocalTechnologyAbility

BEHAVIOUR DEFINED AS:

Indicates the advertised technology ability of the remote hardware. Maps to the Technology Ability Field of the last received Auto-Negotiation Link Code Word(s), defined in clause 28.;

### 30.6.1.1.8 aAutoNegLocalSelectorAbility

ATTRIBUTE

APPROPRIATE SYNTAX:

A SEQUENCE that meets the requirements of the description below:

| | |
|---|---|
| other | Undefined |
| ethernet | IEEE Std 802.3 |
| isoethernet | IEEE Std 802.9 ISLAN-16T |

BEHAVIOUR DEFINED AS:

This indicates the value of the selector field of the local hardware. Selector field is defined in 28.2.1.2.1. The enumeration of the Selector Field indicates the standard that defines the remaining encodings for Auto-Negotiation using that value of enumeration. Additional future enumerations may be assigned to this attribute through the 802.3 maintenance process.;

#### 30.6.1.1.9 aAutoNegAdvertisedSelectorAbility

ATTRIBUTE

APPROPRIATE SYNTAX:
Same as aAutoNegLocalSelectorAbility

BEHAVIOUR DEFINED AS:
This GET-SET attribute maps to the Message Selector Field of the Auto-Negotiation Link Code Word, defined in clause 28. A SET operation to a value not available in aAutoNegLocalSelectorAbility will be rejected. A successful set operation will result in immediate link renegotiation if aAutoNegAdminState is enabled.

NOTE—This will in every case cause temporary link loss during link renegotiation. If set to a value incompatible with aAutoNegReceivedSelectorAbility, link negotiation will not be successful and will cause permanent link loss.;

#### 30.6.1.1.10 aAutoNegReceivedSelectorAbility

ATTRIBUTE

APPROPRIATE SYNTAX:
Same as aAutoNegLocalSelectorAbility

BEHAVIOUR DEFINED AS:
Indicates the advertised message transmission ability of the remote hardware. Maps to the Message Selector Field of the last received Auto-Negotiation Link Code Word, defined in clause 28.;

### 30.6.1.2 Auto-Negotiation actions

#### 30.6.1.2.1 acAutoNegRestartAutoConfig

ATTRIBUTE

APPROPRIATE SYNTAX:
None required

BEHAVIOUR DEFINED AS:
Forces Auto-Negotiation to begin link renegotiation. Has no effect if Auto-Negotiation signaling is disabled.;

#### 30.6.1.2.2 acAutoNegAdminControl

ATTRIBUTE

APPROPRIATE SYNTAX:
Same as aAutoNegAdminState

BEHAVIOUR DEFINED AS:
This action provides a means to turn Auto-Negotiation signaling on or off.;

# Annex 22A

(informative)

# MII output delay, setup, and hold time budget

## 22A.1 System model

The discussion of signal timing characteristics that follows will refer to the system model depicted in figure 22A-1, figure 22A-2, and figure 22A-3. This system model can be applied to each of the three application environments defined in 22.2.1.

Figure 22A-1 depicts a simple system model in which the MII is used to interconnect two integrated circuits on the same circuit assembly. In this model the Reconciliation sublayer comprises one integrated circuit, and the PHY comprises the other. A Reconciliation sublayer or a PHY may actually be composed of several separate integrated circuits. The system model in figure 22A-1 includes two unidirectional signal transmission paths, one from the Reconciliation sublayer to the PHY and one from the PHY to the Reconciliation sublayer. The path from the Reconciliation sublayer to the PHY is separated into two sections, labeled A1 and B1. The path from the PHY to the Reconciliation sublayer is separated into two sections, labeled C1 and D1.

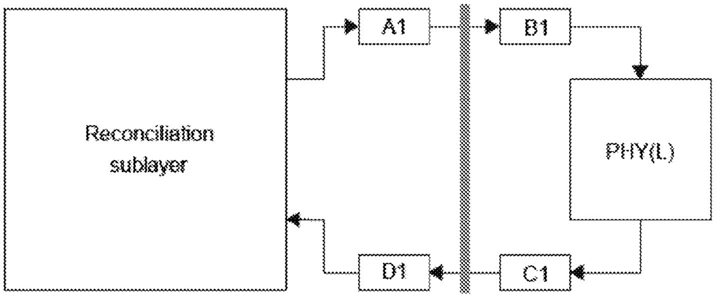


Figure 22A-1—Model for integrated circuit to integrated circuit connection

Figure 22A-1 depicts a system model for the case where the MII is used to interconnect two circuit assemblies. The circuit assemblies may be physically connected in a motherboard/daughterboard arrangement, or they may be physically connected with the cable defined in 22.4.5 and the line interface connector defined in 22.6. The system model in figure 22A-2 includes two unidirectional signal transmission paths, one from the Reconciliation sublayer to the PHY and one from the PHY to the Reconciliation sublayer. The path from the Reconciliation sublayer to the PHY is separated into two sections, labeled A2 and B2. The path from the PHY to the Reconciliation sublayer is separated into two sections, labeled C2 and D2.

Figure 22A-3 depicts a system model in which the MII is used to interconnect both integrated circuits and circuit assemblies. This system model allows for separate signal transmission paths to exist between the Reconciliation sublayer and a local PHY(L), and between the Reconciliation sublayer and a remote PHY(R). The unidirectional paths between the Reconciliation sublayer and the PHY(L) are composed of sections A1, B1, C1, and D1. The unidirectional paths between the Reconciliation sublayer and the remote PHY(R) are composed of sections A2, B2, C2, and D2.

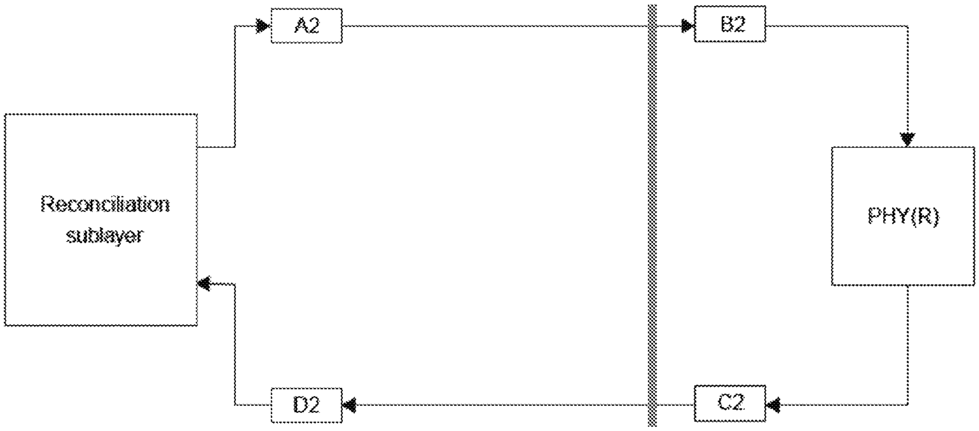


Figure 22A-2—Model for circuit assembly to circuit assembly connection

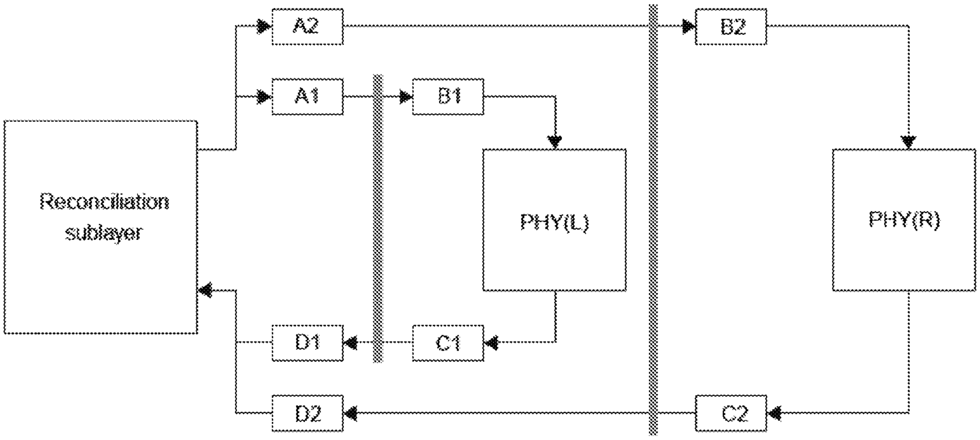


Figure 22A-3—Combined model

Each of these system models assumes a set of common timing and electrical characteristics that shall be met at the input and output ports of the Reconciliation sublayer and PHY devices. The characteristics of the signal transmission paths are identified for each of the sections A1, B1, C1, D1, A2, B2, C2, and D2.

## 22A.2 Signal transmission path characteristics

The signal transmission path characteristics are specified for each of the path sections defined in 22A.1. The characteristics for these sections are specified so as to allow sections A1, B1, C1, and D1 to be implemented in the form of printed circuit board traces, while sections A2, B2, C2, and D2 may be implemented with a combination of printed circuit board traces and wire conductors in a cable assembly.

The signal transmission path characteristics are stated in terms of their maximum delay and their characteristic impedance. These values are summarized in table 22A-1.

**Table 22A-1—Signal transmission path characteristics**

| Section | Maximum delay (ns) | Impedance (Ω) |
|---|---|---|
| A1, D1 | 5 | 68 ± 15% |
| B1, C1 | 2.5 | 68 ± 15% |
| A2, D2 | 5 | 68 ± 10% |
| B2, C2 | 2.5 | 68 ± 10% |

The driver characteristics specified in 22.4.3, the receiver characteristics specified in 22.4.4, and the signal transmission path characteristics specified in table 22A-1 can be applied to the system models shown in figure 22A-1 or figure 22A-2. The combination of loads presented in figure 22A-3 cannot be adequately driven by an output buffer that meets the driver characteristics specified in 22.4.3 while being sampled by an input buffer that meets the receiver characteristics specified in 22.4.4.

To address the system model depicted in figure 22A-3, it is permissible to incorporate an additional stage of buffering into path sections A1, A2, D1, and D2, provided that the resulting maximum delay characteristic for those path sections does not exceed the value stated in table 22A-1. The delay characteristic for transmission path sections A2 and D2 includes an allowance for the delay that results from the presence of a lumped capacitive load at the end of the path. For a transmission path section with a characteristic impedance $Z_0$, with a lumped capacitive load $C_L$, this delay is nominally $Z_0C_L$. In the case of a maximum transmission path section impedance of 78 Ω with a lumped load of 8 pF, the nominal delay is 0.6 ns. Thus the allowable delay for a buffer inserted into transmission path section A2 or D2 is 4.4 ns.

## 22A.3 Budget calculation

A recommended timing budget is shown in table 22A-2. This budget assumes that the combined system model shown in figure 22A-3 represents a worst case.

**Table 22A-2—Round-trip delay budget**

| Description | Incremental delay (ns) | Cumulative delay (ns) |
|---|---|---|
| TX_CLK output at PHY(R) | 0.0 | 0.0 |
| Transmission path section C2 | 2.5 | 2.5 |
| Transmission path section D2 | 5.0 | 7.5 |
| clock to output in Reconciliation sublayer | 15.0 | 22.5 |
| Transmission path section A2 | 5.0 | 27.5 |
| Transmission path section B2 | 2.5 | 30.0 |
| Setup time at PHY(R) | 10.0 | 40.0 |

# Annex 22B

(informative)

# MII driver ac characteristics

## 22B.1 Implications of CMOS ASIC processes

For MII drivers that drive rail to rail, such as those commonly used in CMOS ASICs (complimentary metal oxide semiconductor application-specific integrated circuits), the ac characteristic performance requirements of 22.4.3.2 can be met if the $V_{oh}$ vs. $I_{oh}$ and $V_{ol}$ vs. $I_{ol}$ dc characteristics of the driver stay within the unshaded areas of figure 22B-1.

The variation in output resistance of a field effect transistor (FET) due to variations in supply voltage, temperature, and process may require that a resistance be placed in series with the output of the FETs to meet this specification. The series resistance can be part of the driver circuit, or external to the driver. If the series resistance is not part of the driver circuit, the driver vendor shall specify the value of series resistance required to meet the specification. A series resistor used to meet this specification is conceptually part of the driver regardless of whether it is physically internal or external to the driver.

The propagation delay of the path between the driver and an external series resistor used to meet the specification shall not exceed 10% of the 10–90% rise/fall time of the driver.

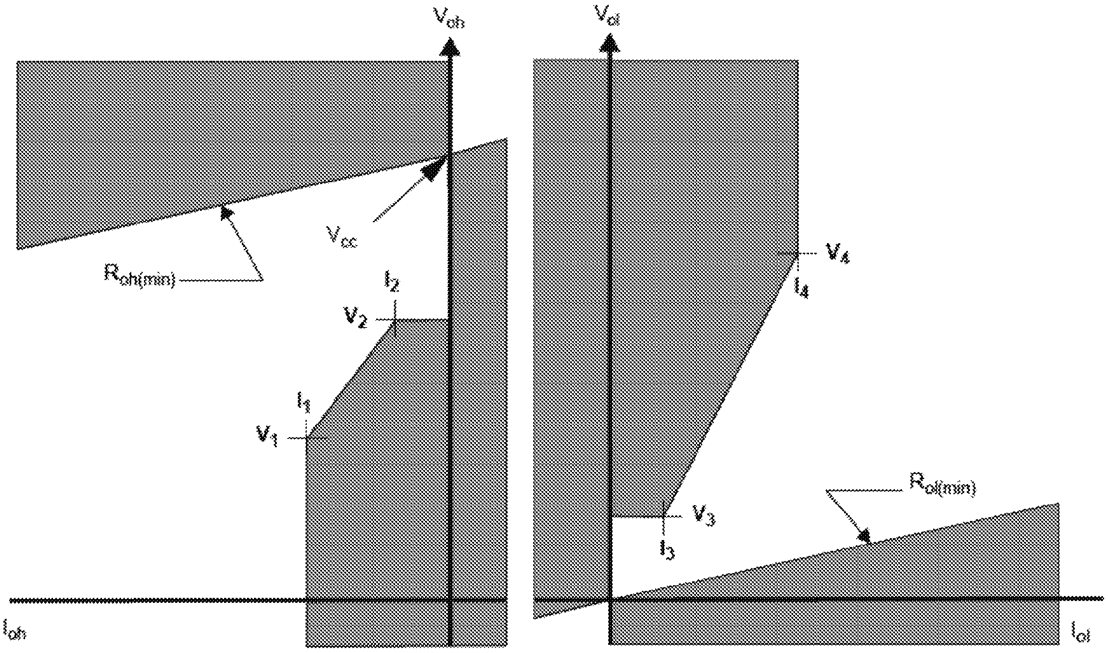


Figure 22B-1—Driver output V–I curve

## 22B.2 $R_{o(min)}$ and V, I values for operation from 5 V ±10% supply

Referring to figure 22B-1, $R_{oh(min)}$ and $R_{ol(min)}$ both equal 40 | , and the values for the V-I points on the curve are given in table 22B-1 below for MII drivers that drive rail to rail from a +5 V ± 10% power supply.

**Table 22B-1—Values for driver output V-I curve (5 V supply)**

| V–I point | I (mA) | V (V) |
|---|---|---|
| $I_1$, $V_1$ | –20 | 1.10 |
| $I_2$, $V_2$ | –4 | 2.4 |
| $I_3$, $V_3$ | 4 | 0.40 |
| $I_4$, $V_4$ | 43 | 3.05 |

## 22B.3 $R_{o(min)}$ and V, I values for operation from 3.3 ± 0.3 V supply

Referring to figure 22B-1, $R_{oh(min)}$ and $R_{ol(min)}$ both equal 33 | , and the values for the V–I points on the curve are given in table 22B-2 below for MII drivers that drive rail to rail from a +3.3 ± 0.3 V power supply.

**Table 22B-2—Values for driver output V-I curve (3.3 V supply)**

| V-I point | I (mA) | V (V) |
|---|---|---|
| $I_1$,$V_1$ | –20 | 1.10 |
| $I_2$,$V_2$ | –4 | 2.4 |
| $I_3$,$V_3$ | 4 | 0.40 |
| $I_4$,$V_4$ | 26 | 2.10 |

# Annex 22C

(informative)

# Measurement techniques for MII signal timing characteristics

## 22C.1 Measuring timing characteristics of source terminated signals

The measurement of timing relationships between MII signals at the MII connector is complicated by the use of driver output impedance to control transmission line reflections on point-to-point transmission paths passing through the connector. The voltage waveforms on point-to-point transmission paths can be different at the MII connector and at the end of the paths. A clean transition (or step) from one logic state to the other at the end of a point to point path can appear as two half-steps at the MII connector.

To eliminate ambiguity as to where on a two half-step state transition to measure timing, all timing measurements on point-to-point transmission paths will be at the end of the path. In some cases, an end of path must be artificially created.

## 22C.2 Measuring timing characteristics of transmit signals at the MII

The timing of TX_EN, TX_ER, and TXD<3:0> relative to TX_CLK at the MII connector is measured as follows.

Use the time base for TX_CLK as a timing reference. Break the TX_CLK path at the MII connector, forcing the TX_CLK point-to-point transmission path to end at the connector. Measure when the rising edge of TX_CLK passes through $V_{ih(min)}$ at the MII connector. Call this time $T_{clk}$. Reconnect the TX_CLK path at the MII connector and break the paths of TX_EN, TX_ER, and TXD<3:0> at the MII connector, forcing the paths to end at the connector. Measure when TX_EN, TX_ER, and TXD<3:0> exit the switching region at the MII connector. Call these times $T_{en}$, $T_{er}$, and $T_{<3:0>}$, respectively.

The timing relationships at the MII connector for TX_EN, TX_ER, and TXD<3:0> relative to TX_CLK are met if $(T_{en} - T_{clk})$, $(T_{er} - T_{clk})$, $(T_3 - T_{clk})$, $(T_2 - T_{clk})$, $(T_1 - T_{clk})$, and $(T_0 - T_{clk})$, respectively, meet the timing relationships specified in 22.3.1.

## 22C.3 Measuring timing characteristics of receive signals at the MII

The timing of RX_DV, RX_ER, and RXD<3:0> relative to RX_CLK at the MII connector is measured as follows.

Break the paths of RX_CLK, RX_DV, RX_ER, and RXD<3:0> at the MII connector, forcing the paths to end at the connector. Measure when RX_DV, RX_ER, and RXD<3:0> exit the switching region at the MII connector relative to when the rising edge of RX_CLK passes through $V_{il(max)}$. Also measure when RX_DV, RX_ER, and RXD<3:0> reenter the switching region relative to when the rising edge of RX_CLK passes through $V_{ih(min)}$.

The timing relationships at the MII connector for RX_DV, RX_ER, and RXD<3:0> relative to RX_CLK are met if the times measured in the previous step meet the timing relationships specified in 22.3.2.

## 22C.4 Measuring timing characteristics of MDIO

The MDIO and MDC signal timing characteristics cannot be measured using the techniques defined for the transmit and receive signals since MDIO and MDC may connect a single station management entity to multiple PHY devices. The MDIO and MDC timing characteristics are measured with a PHY connected to the MII connector. The signal timing characteristics for MDC and MDIO must be met over the range of conditions which occur when from one to 32 PHYs are connected to an STA. When 32 PHYs are connected to an STA, the total capacitance can be as large as 390 pF on MDC, and as large as 470 pF on MDIO.

# Annex 23A

(normative)

## 6T code words

The *leftmost* ternary symbol of each 6T Code group shown in table 23A-1 (broken into 23A-1a and 23A-1b for pagination) shall be transmitted *first.* The *leftmost* nibble of each data octet is the *most significant.*

**Table 23A-1a—100BASE-T4 8B6T code table**

| Data octet | 6T code group | Data octet | 6T code group | Data octet | 6T code group | Data octet | 6T code group |
|---|---|---|---|---|---|---|---|
| 00 | + − 0 0 + − | 20 | 0 0 − + + − | 40 | + 0 + 0 0 − | 60 | 0 − 0 + + 0 |
| 01 | 0 + − + − 0 | 21 | − − + 0 0 + | 41 | + + 0 0 − 0 | 61 | 0 0 − + 0 + |
| 02 | + − 0 + − 0 | 22 | + + − 0 + − | 42 | + 0 + 0 − 0 | 62 | 0 − 0 + 0 + |
| 03 | − 0 + + − 0 | 23 | + + − 0 − + | 43 | 0 + + 0 − 0 | 63 | − 0 0 + 0 + |
| 04 | − 0 + 0 + − | 24 | 0 0 + 0 − + | 44 | 0 + + 0 0 − | 64 | − 0 0 + + 0 |
| 05 | 0 + − − 0 + | 25 | 0 0 + 0 + − | 45 | + + 0 − 0 0 | 65 | 0 0 − 0 + + |
| 06 | + − 0 − 0 + | 26 | 0 0 − 0 0 + | 46 | + 0 + − 0 0 | 66 | 0 − 0 0 + + |
| 07 | − 0 + − 0 + | 27 | − − + + + − | 47 | 0 + + − 0 0 | 67 | − 0 0 0 + + |
| 08 | − + 0 0 + − | 28 | − 0 − + + 0 | 48 | 0 0 0 + 0 0 | 68 | − + − + + 0 |
| 09 | 0 − + + − 0 | 29 | − − 0 + 0 + | 49 | 0 0 0 − + + | 69 | − − + + 0 + |
| 0A | − + 0 + − 0 | 2A | − 0 − + 0 + | 4A | 0 0 0 + − + | 6A | − + − + 0 + |
| 0B | + 0 − + − 0 | 2B | 0 − − + 0 + | 4B | 0 0 0 + + − | 6B | + − − + 0 + |
| 0C | + 0 − 0 + − | 2C | 0 − − + + 0 | 4C | 0 0 0 − + 0 | 6C | + − − + + 0 |
| 0D | 0 − + − 0 + | 2D | − − 0 0 + + | 4D | 0 0 0 − 0 + | 6D | − − + 0 + + |
| 0E | − + 0 − 0 + | 2E | − 0 − 0 + + | 4E | 0 0 0 + − 0 | 6E | − + − 0 + + |
| 0F | + 0 − − 0 + | 2F | 0 − − 0 + + | 4F | 0 0 0 + 0 − | 6F | + − − 0 + + |
| 10 | + 0 + − − 0 | 30 | + − 0 0 − + | 50 | + 0 + − − + | 70 | − + + 0 0 0 |
| 11 | + + 0 − 0 − | 31 | 0 + − − + 0 | 51 | + + 0 − + − | 71 | + − + 0 0 0 |
| 12 | + 0 + − 0 − | 32 | + − 0 − + 0 | 52 | + 0 + − + − | 72 | + + − 0 0 0 |
| 13 | 0 + + − 0 − | 33 | − 0 + − + 0 | 53 | 0 + + − + − | 73 | 0 0 + 0 0 0 |
| 14 | 0 + + − − 0 | 34 | − 0 + 0 − + | 54 | 0 + + − − + | 74 | − 0 + 0 0 0 |
| 15 | + + 0 0 − − | 35 | 0 + − + 0 − | 55 | + + 0 + − − | 75 | 0 − + 0 0 0 |
| 16 | + 0 + 0 − − | 36 | + − 0 + 0 − | 56 | + 0 + + − − | 76 | + 0 − 0 0 0 |
| 17 | 0 + + 0 − − | 37 | − 0 + + 0 − | 57 | 0 + + + − − | 77 | 0 + − 0 0 0 |
| 18 | 0 + − 0 + − | 38 | − + 0 0 − + | 58 | + + + 0 − − | 78 | 0 − − + + + |
| 19 | 0 + − 0 − + | 39 | 0 − + − + 0 | 59 | + + + − 0 − | 79 | − 0 − + + + |
| 1A | 0 + − + + − | 3A | − + 0 − + 0 | 5A | + + + − − 0 | 7A | − − 0 + + + |
| 1B | 0 + − 0 0 + | 3B | + 0 − − + 0 | 5B | + + 0 − − 0 | 7B | − − 0 + + 0 |
| 1C | 0 − + 0 0 + | 3C | + 0 − 0 − + | 5C | + + 0 − − + | 7C | + + − 0 0 − |
| 1D | 0 − + + + − | 3D | 0 − + + 0 − | 5D | + + 0 0 0 − | 7D | 0 0 + 0 0 − |
| 1E | 0 − + 0 − + | 3E | − + 0 + 0 − | 5E | − − + + + 0 | 7E | + + − − − + |
| 1F | 0 − + 0 + − | 3F | + 0 − + 0 − | 5F | 0 0 − + + 0 | 7F | 0 0 + − − + |

### Table 23A-1b—100BASE-T4 8B6T code table

| Data octet | 6T code group | Data octet | 6T code group | Data octet | 6T code group | Data octet | 6T code group |
|---|---|---|---|---|---|---|---|
| 80 | + − + 0 0 − | A0 | 0 − 0 + + − | C0 | + − + 0 + − | E0 | + − 0 + + − |
| 81 | + + − 0 − 0 | A1 | 0 0 − + − + | C1 | + + − + − 0 | E1 | 0 + − + − + |
| 82 | + − + 0 − 0 | A2 | 0 − 0 + − + | C2 | + − + + − 0 | E2 | + − 0 + − + |
| 83 | − + + 0 − 0 | A3 | − 0 0 + − + | C3 | − + + + − 0 | E3 | − 0 + + − + |
| 84 | − + + 0 0 − | A4 | − 0 0 + + − | C4 | − + + 0 + − | E4 | − 0 + + + − |
| 85 | + + − − 0 0 | A5 | 0 0 − − + + | C5 | + + − − 0 + | E5 | 0 + − − + + |
| 86 | + − + − 0 0 | A6 | 0 − 0 − + + | C6 | + − + − 0 + | E6 | + − 0 − + + |
| 87 | − + + − 0 0 | A7 | − 0 0 − + + | C7 | − + + − 0 + | E7 | − 0 + − + + |
| 88 | 0 + 0 0 0 − | A8 | − + − + + − | C8 | 0 + 0 0 + − | E8 | − + 0 + + − |
| 89 | 0 0 + 0 − 0 | A9 | − − + + − + | C9 | 0 0 + + − 0 | E9 | 0 − + + − + |
| 8A | 0 + 0 0 − 0 | AA | − + − + − + | CA | 0 + 0 + − 0 | EA | − + 0 + − + |
| 8B | + 0 0 0 − 0 | AB | + − − + − + | CB | + 0 0 + − 0 | EB | + 0 − + − + |
| 8C | + 0 0 0 0 − | AC | + − − + + − | CC | + 0 0 0 + − | EC | + 0 − + + − |
| 8D | 0 0 + − 0 0 | AD | − − + − + + | CD | 0 0 + − 0 + | ED | 0 − + − + + |
| 8E | 0 + 0 − 0 0 | AE | − + − − + + | CE | 0 + 0 − 0 + | EE | − + 0 − + + |
| 8F | + 0 0 − 0 0 | AF | + − − − + + | CF | + 0 0 − 0 + | EF | + 0 − − + + |
| 90 | + − + − − + | B0 | 0 − 0 0 0 + | D0 | + − + 0 − + | F0 | + − 0 0 0 + |
| 91 | + + − − + − | B1 | 0 0 − 0 + 0 | D1 | + + − − + 0 | F1 | 0 + − 0 + 0 |
| 92 | + − + − + − | B2 | 0 − 0 0 + 0 | D2 | + − + − + 0 | F2 | + − 0 0 + 0 |
| 93 | − + + − + − | B3 | − 0 0 0 + 0 | D3 | − + + − + 0 | F3 | − 0 + 0 + 0 |
| 94 | − + + − − + | B4 | − 0 0 0 0 + | D4 | − + + 0 − + | F4 | − 0 + 0 0 + |
| 95 | + + − + − − | B5 | 0 0 − + 0 0 | D5 | + + − + 0 − | F5 | 0 + − + 0 0 |
| 96 | + − + + − − | B6 | 0 − 0 + 0 0 | D6 | + − + + 0 − | F6 | + − 0 + 0 0 |
| 97 | − + + + − − | B7 | − 0 0 + 0 0 | D7 | − + + + 0 − | F7 | − 0 + + 0 0 |
| 98 | 0 + 0 − − + | B8 | − + − 0 0 + | D8 | 0 + 0 0 − + | F8 | − + 0 0 0 + |
| 99 | 0 0 + − + − | B9 | − − + 0 + 0 | D9 | 0 0 + − + 0 | F9 | 0 − + 0 + 0 |
| 9A | 0 + 0 − + − | BA | − + − 0 + 0 | DA | 0 + 0 − + 0 | FA | − + 0 0 + 0 |
| 9B | + 0 0 − + − | BB | + − − 0 + 0 | DB | + 0 0 − + 0 | FB | + 0 − 0 + 0 |
| 9C | + 0 0 − − + | BC | + − − 0 0 + | DC | + 0 0 0 − + | FC | + 0 − 0 0 + |
| 9D | 0 0 + + − − | BD | − − + + 0 0 | DD | 0 0 + + 0 − | FD | 0 − + + 0 0 |
| 9E | 0 + 0 + − − | BE | − + − + 0 0 | DE | 0 + 0 + 0 − | FE | − + 0 + 0 0 |
| 9F | + 0 0 + − − | BF | + − − + 0 0 | DF | + 0 0 + 0 − | FF | + 0 − + 0 0 |

# Annex 23B

(informative)

# Noise budget

Worst-case values for noise effects in the 100BASE-T4 system are as shown in tables 23B-1 and 23B-2.

**Table 23B-1—Carrier presence analysis**

| | |
|---|---|
| Received signal peak amplitude (min.) | 792 mVp |
| NEXT noise | 325 mVp |

**Table 23B-2—Far-end signal analysis**

| | |
|---|---|
| Received signal peak amplitude (min.) | 796 mVp |
| Baseline wander | 14 mVp |
| ISI | 80 mVp |
| Reflections | 60 mVp |
| FEXT noise | 87 mVp |

# Annex 23C

(informative)

## Use of cabling systems with a nominal differential characteristic impedance of 120 |

The 100BASE-T4 standard specifies only the use of 100 | link segments for conformance. Since ISO/IEC 11801: 1995 also recognizes 120 | cabling, this informative annex specifies the conditions for using cabling systems with a nominal characteristic impedance of 120 | by 100BASE-T4 conformant stations.

The use of cables with a characteristic impedance outside the range specified in 23.6 will generally increase the mismatching effects in the link components, inducing additional noise in the received signals.

In particular, the use of a homogeneous link segment having a characteristic impedance of 120 | ±15 | over the frequency band 1 to 16 MHz may add up to 1.4% of additional noise to the signals at the input of the receivers (worst-case short-length link segment).

Therefore, in order to keep the overall noise (MDFEXT + reflections) at the same value as for a 100 | link segment when using a 120 | link segment, the minimum ELFEXT loss requirement for the cable must be increased by 2 dB (i.e., from 23 dB to 25 dB at 12.5 MHz, see 23.6.3.2). Accordingly, the MDFEXT noise requirement shall be decreased from 87 mV peak to 69 mV peak. In practice, this means that cables rated category 4 or higher, as specified in ISO/IEC 11801: 1995, are required when 120 | cables are used with 100BASE-T4 compliant PMDs.

NOTES

1—The use of 100 | cords at end points in conjunction with 120 | premises cabling may be tolerated provided that all the components of the link are of category 5, as defined in ISO/IEC 11801: 1995.

2—The use of 100 | cords at any intermediate cross-connect points on 120 | links as well as the use of 120 | cords in conjunction with 100 | premises cabling is not allowed since it would result in worst-case jitter greater than that allowed in this standard.

CAUTION—Users of this annex are further advised to check with the manufacturer of the particular 100BASE-T4 couplers they intend to use with a 120 | link to see whether those couplers can operate correctly on cables with $Z_c$ as high as 120 | ± 15 | .

# Annex 27A

(normative)

## Repeater delay consistency requirements

Proper operation of the network requires that repeaters do not cause the Inter-Packet Gap (IPG) to disappear by propagating the end of any carrier event to different output ports with greatly different delay times. Maximum port-to-port delays have been assigned as absolute delays to meet requirements for detection of collision within a slot time and limiting the length of collision fragments to less than minimum frame size. To avoid specification of minimum input-to-output propagation time as absolute values that reduce implementation flexibility, these delays are instead implied by imposing a triangular delay inequality relationship.

Consider three ports {A, B, C}. Using the notation SOP(xy) to mean the start-of-packet delay for an input at port x to resulting output on port y, repeaters shall achieve this relationship for all groups of three ports within a repeater set:

$$\mathrm{SOP(AC)} < \mathrm{SOP(AB)} + \mathrm{SOP(BC)}$$

Following a frame transmitted by node A that propagates to nodes B and C, this constraint ensures that node B cannot complete an IPG timer and initiate a transmission that arrives at node C before node C has also advanced its own IPG timer sufficiently that a pending frame can contend for access to the network.

There is a second delay consistency requirement, one that relates to jam propagation by repeaters. Using a notation similar to that above, SOJ(xy) stands for the start-of-jam propagation delay from port x to port y and EOJ(xy) for the end-of-jam delay between same two ports.

To ensure proper detection of collisions and avoid generation of fragments that exceed minimum frame size, maximum values have been imposed on SOJ and EOJ delays through repeaters. No specific minima have been specified as all delays less than the maxima meet the collision detection and fragment length criteria. To prevent the jam pattern from shrinking excessively as it propagates through repeaters, repeaters shall meet this relationship between all pairs of ports:

$$\mathrm{EOJ(AB)} >= \mathrm{SOJ(AB)} - 4 \text{ bit times}$$

# Annex 28A

(normative)

# Selector Field definitions

The Selector Field, S[4:0] in the Link Code Word, shall be used to identify the type of message being sent by Auto-Negotiation. The following table identifies the types of messages that may be sent. As new messages are developed, this table will be updated accordingly.

The Selector Field uses a 5-bit binary encoding, which allows 32 messages to be defined. All unspecified combinations are reserved. Reserved combinations shall not be transmitted.

**Table 28A-1—Selector Field value mappings**

| S4 | S3 | S2 | S1 | S0 | Selector description |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | Reserved for future Auto-Negotiation development |
| 0 | 0 | 0 | 0 | 1 | IEEE Std 802.3 |
| 0 | 0 | 0 | 1 | 0 | IEEE Std 802.9 ISLAN-16T |
| 1 | 1 | 1 | 1 | 1 | Reserved for future Auto-Negotiation development |

# Annex 28B

(normative)

# IEEE 802.3 Selector Base Page definition

This annex provides the Technology Ability Field bit assignments, Priority Resolution table, and Message Page transmission conventions relative to the IEEE 802.3 Selector Field value within the base page encoding.

As new IEEE 802.3 LAN technologies are developed, a reserved bit in the Technology Ability Field may be assigned to each technology by the standards body.

The new technology will then be inserted into the Priority Resolution hierarchy and made a part of the Auto-Negotiation standard. The relative hierarchy of the existing technologies will not change, thus providing backward compatibility with existing Auto-Negotiation implementations.

It is important to note that the reserved bits are required to be transmitted as logic zeros. This guarantees that devices implemented using the current priority table will be forward compatible with future devices using an updated priority table.

## 28B.1 Selector field value

The value of the IEEE 802.3 Selector Field is S[4:0] = 00001.

## 28B.2 Technology Ability Field bit assignments

The Technology bit field consists of bits D5 through D12 (A0–A8 respectively) in the IEEE 802.3 Selector Base Page. Table 28B-1 summarizes the bit assignments.

Note that the order of the bits within the Technology Ability Field has no relationship to the relative priority of the technologies.

**Table 28B-1—Technology Ability Field bit assignments**

| Bit | Technology | Minimum cabling requirement |
|---|---|---|
| A0 | 10BASE-T | Two-pair Category 3 |
| A1 | 10BASE-T full duplex | Two-pair Category 3 |
| A2 | 100BASE-TX | Two-pair Category 5 |
| A3 | 100BASE-TX full duplex | Two-pair Category 5 |
| A4 | 100BASE-T4 | Four-pair Category 3 |
| A5 | Reserved for future technology | |
| A6 | Reserved for future technology | |
| A7 | Reserved for future technology | |

## 28B.3 Priority resolution

Since two devices may have multiple abilities in common, a prioritization scheme exists to ensure that the highest common denominator ability is chosen. The following list shall represent the relative priorities of the technologies supported by the IEEE 802.3 Selector Field value, where priorities are listed from highest to lowest.

a) 100BASE-TX full duplex
b) 100BASE-T4
c) 100BASE-TX
d) 10BASE-T full duplex
e) 10BASE-T

The rational for this hierarchy is straightforward. 10BASE-T is the lowest common denominator and therefore has the lowest priority. Full-duplex solutions are always higher in priority than their half-duplex counterparts. 100BASE-T4 is ahead of 100BASE-TX because 100BASE-T4 runs across a broader spectrum of copper cabling. The relative order of the technologies specified herein shall not be changed. As each new technology is added, it shall be inserted into its appropriate place in the list, shifting technologies of lesser priority lower in priority. If a vendor-specific technology is implemented, the priority of all IEEE 802.3 standard technologies shall be maintained, with the vendor-specific technology inserted at any appropriate priority location.

## 28B.4 Message Page transmission convention

Each series of Unformatted Pages shall be preceded by a Message Page containing a Message Code that defines how the following Unformatted Pages will be used.

Next Page message codes should be allocated globally across Selector Field values so that meaningful communication is possible between technologies using different Selector Field values.

# Annex 28C

(normative)

## Next Page Message Code Field definitions

The Message Code Field of a message page used in Next Page exchange shall be used to identify the meaning of a message. The following table identifies the types of messages that may be sent. As new messages are developed, this table will be updated accordingly.

The Message Code Field uses an 11-bit binary encoding that allows 2048 messages to be defined. All Message Codes not specified shall be reserved for IEEE use or allocation.

Table 28C-1—Message Code Field values

| Message code # | M10 | M9 | M8 | M7 | M6 | M5 | M4 | M3 | M2 | M1 | M0 | Message Code description |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Reserved for future Auto-Negotiation use |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Null Message |
| 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | One UP with Technology Ability Field follows |
| 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Two UPs with Technology Ability Field follows |
| 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | One UP with Binary coded Remote fault follows |
| 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | Organizationally Unique Identifier Tagged Message |
| 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | PHY Identifier Tag Code |
| 2047 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | Reserved for future Auto-Negotiation use |

### 28C.1 Message code #0—Auto-Negotiation reserved code 1

This code is reserved for future Auto-Negotiation function enhancements. Devices shall not transmit this code.

### 28C.2 Message code #1—Null Message code

The Null Message code shall be transmitted during Next Page exchange when the Local Device has no further messages to transmit and the Link Partner is still transmitting valid Next Pages. See 28.2.3.4 for more details.

## 28C.3 Message code #2—Technology Ability extension code 1

This Message Code is reserved for future expansion of the Technology Ability Field and indicates that a defined user code with a specific Technology Ability Field encoding follows.

## 28C.4 Message code #3—Technology Ability extension code 2

This Message Code is reserved for future expansion of the Technology Ability Field and indicates that two defined user codes with specific Technology Ability Field encodings follow.

## 28C.5 Message code #4—Remote fault number code

This Message Code shall be followed by a single user code whose encoding specifies the type of fault that has occurred. The following user codes are defined:

0: RF Test
    This code can be used to test Remote Fault operation.
1: Link Loss
2: Jabber
3: Parallel Detection Fault
    This code may be sent to identify when bit 6.4 is set.

## 28C.6 Message code #5—Organizationally Unique Identifier (OUI) tag code

The OUI Tagged Message shall consist of a single message code of 0000 0000 0101 followed by four user codes defined as follows. The first user code shall contain the most significant 11 bits of the OUI (bits 23:13) with the most significant bit in bit 10 of the user code. The second user code shall contain the next most significant 11 bits of the OUI (bits 12:2) with the most significant bit in bit 10 of the user code. The third user code shall contain the remaining least significant 2 bits of the OUI (bits 1:0) with the most significant bit in bit 10 of the user code. Bits 8:0 of the fourth user contain a user-defined user code value that is specific to the OUI transmitted. The fourth and final user code shall contain a user-defined user code value that is specific to the OUI transmitted.

## 28C.7 Message code #6—PHY identifier tag code

The PHY ID tag code message shall consist of a single message code of 0000 0000 0110 followed by four user codes defined as follows. The first user code shall contain the most significant 11 bits of the PHY ID (2.15:5) with the most significant bit in bit 10 of the user code. The second user code shall contain bits 2.4:0 to 3.15:10 of the PHY ID with the most significant bit in bit 10 of the user code. The third user code shall contain bits 3.9:0 of the PHY ID with the most significant bit in bit 10 of the user code. Bit 0 in the third user code shall contain a user-defined user code value that is specific to the PHY ID transmitted. The fourth and final user code shall contain a user-defined user code value that is specific to the PHY ID transmitted.

## 28C.8 Message code #2047—Auto-Negotiation reserved code 2

This code is reserved for future Auto-Negotiation function enhancements. Devices shall not transmit this code.

# Annex 29A

(informative)

# DTE and repeater delay components

## 29A.1 DTE delay

Round-trip DTE delay = MAC transmit start to MDI output
+ MDI input to MDI output (worst case, nondeferred)
+ MDI input to collision detect

NOTES

1—Refer to clauses 23, 24, 25, and 26.

2—Worst-case values are used for the one T4 and one TX/FX value shown in table 29-3. (TX/FX values for MAC transmit start and MDI input to collision detect; T4 value for MDI input to MDI output.)

## 29A.2 Repeater delay

Repeater delay= SOP (start-of-packet propagation delay)
+ SOJ (start-of-jam propagation delay)

NOTE—Refer to clause 27.

# Annex 29B

(informative)

# Recommended topology documentation

It is strongly recommended that detailed records documenting the topology components of 100BASE-T networks be prepared and maintained to facilitate subsequent modification. Proper 100BASE-T topology design requires an accurate knowledge of link segment and hub parameters to ensure proper operation of single and multi-segment, single collision domain networks. Link segment documentation is site-specific and requires careful documentation. It is recommended that the information shown in table 29B-1 be collected for each link segment and archived for future reference. Hub performance parameters may be obtained from manufacturer documentation.

**Table 29B-1—Recommended link segment documentation**

| | Horizontal wiring (wiring closet, from punch-down block to end station wall plate) | MII cable(s) | Wiring closet patch cord | End station connecting cable |
|---|---|---|---|---|
| Length | | | | |
| Type (e.g., Category 3) | | | | |
| Cable manufacturer | | | | |
| Cable code/id (from manufacturer) | | | | |
| Cable delay (in bit times per meter) | | | | |

# Annex 30A

(normative)

# GDMO specification for 802.3 managed object classes

This annex formally defines the protocol encodings for CMIP and ISO/IEC 15802-2: 1995 [IEEE 802.1B] for the IEEE 802.3 Managed Objects using the templates specified in ISO/IEC 10165-4: 1992, Guidelines for the definition of managed objects (GDMO). The application of a GDMO template compiler against 30A.1 to 30A.8 will produce the proper protocol encodings.

NOTE—The arcs (that is, object identifier values) defined in annex 30A deprecate the arcs previously defined in Annexes D1 (Layer Management), D2 (Repeater Management), and D3 (MAU Management). See IEEE Std 802.1F-1993, annex C.4.

Each attribute definition in this clause references directly by means of the WITH ATTRIBUTE SYNTAX construct or indirectly by means of the DERIVED FROM construct an ASN.1 type or subtype that defines the attribute's type and range. Those ASN.1 types and subtypes defined exclusively for CSMA/CD Management appear in a single ASN.1 module at the end of this annex.

Counters for these protocol encodings are specified as either 32 or 64 bits wide. Thirty-two bit counters are used for the protocol encoding of counter attributes, providing the minimum rollover time is 58 min or more. Sixty-four bit counters are used for the protocol encoding of counter attributes that could roll over in less than 58 min with a 32-bit counter. Approximate counter rollover times are provided as notes below each counter BEHAVIOUR definition. Approximate rollover time for 100 Mb/s operation is one tenth the value of the approximate rollover time for 10 Mb/s operation except where indicated, or where one tenth the value for 10 Mb/s operation is less than 58 min. For formal definition of the counter, refer to the BEHAVIOUR bCMCounter in 30B.1.

## 30A.1 DTE MAC entity managed object class

### 30A.1.1 DTE MAC entity formal definition

```
oMACEntity                              MANAGED OBJECT CLASS

    DERIVED FROM                        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
                                        1992":top;
    CHARACTERIZED BY
        pBasic                          PACKAGE
            ATTRIBUTES                  aMACID                      GET;
            ACTIONS                     acInitializeMAC;
        ;
    ;
    CONDITIONAL PACKAGES
        pMandatory                      PACKAGE
            ATTRIBUTES                  aFramesTransmittedOK        GET,
                                        aSingleCollisionFrames      GET,
                                        aMultipleCollisionFrames    GET,
                                        aFramesReceivedOK           GET,
                                        aFrameCheckSequenceErrors   GET,
                                        aAlignmentErrors            GET;
            REGISTERED AS               {iso(1) member-body(2) us(840) 802dot3(10006)
```

```
                                        csmacdmgt(30) package(4) macMandatoryPkg(1)};
            PRESENT IF                  Conformance to DTE Management is desired.;
        pRecommended                    PACKAGE
            ATTRIBUTES                  aOctetsTransmittedOK              GET,
                                        aFramesWithDeferredXmissions      GET,
                                        aLateCollisions                   GET,
                                        aFramesAbortedDueToXSColls        GET,
                                        aFramesLostDueToIntMACXmitError       GET,
                                        aCarrierSenseErrors                   GET,
                                        aOctetsReceivedOK                     GET,
                                        aFramesLostDueToIntMACRcvError        GET,
                                        aPromiscuousStatus                    GET-SET,
                                        aReadMulticastAddressList             GET;
            ACTIONS                     acAddGroupAddress,
                                        acDeleteGroupAddress;
            REGISTERED AS               {iso(1) member-body(2) us(840) 802dot3(10006)
                                        csmacdmgt(30) package(4)
                                        macRecommendedPkg(2)};
            PRESENT IF                  The Recommended Package is implemented.;
        pOptional                       PACKAGE
            ATTRIBUTES                  aMulticastFramesXmittedOK         GET,
                                        aBroadcastFramesXmittedOK         GET,
                                        aMulticastFramesReceivedOK        GET,
                                        aBroadcastFramesReceivedOK        GET,
                                        aInRangeLengthErrors              GET,
                                        aOutOfRangeLengthField            GET,
                                        aFrameTooLongErrors               GET,
                                        aMACEnableStatus                  GET-SET,
                                        aTransmitEnableStatus             GET-SET,
                                        aMulticastReceiveStatus           GET-SET,
                                        aReadWriteMACAddress              GET-SET;
            ACTIONS                     acExecuteSelfTest;
            REGISTERED AS               {iso(1) member-body(2) us(840) 802dot3(10006)
                                        csmacdmgt(30) package(4) optionalPkg(3)};
            PRESENT IF                  The Optional Package and the Recommended Package
                                        are implemented.;
        pArray                          PACKAGE
            ATTRIBUTES                  aCollisionFrames                  GET;
            REGISTERED AS               {iso(1) member-body(2) us(840) 802dot3(10006)
                                        csmacdmgt(30) package(4) arrayPkg(4)};
            PRESENT IF                  The Array Package and the Recommended Package
                                        are implemented.;
        pExcessiveDeferral              PACKAGE
            ATTRIBUTES                  aFramesWithExcessiveDeferral      GET;
            REGISTERED AS               {iso(1) member-body(2) us(840) 802dot3(10006)
                                        csmacdmgt(30) package(4)
                                        excessiveDeferralPkg(5)};
            PRESENT IF                  The ExcessiveDeferral Package and the
                                        Recommended Package are implemented.;
    REGISTERED AS           {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                            managedObjectClass(3) macObjectClass(1)};


nbMACName                               NAME BINDING
```

SUBORDINATE OBJECT CLASS oMACEntity;
NAMED BY SUPERIOR OBJECT CLASS
"ISO/IEC 10165-2":system;
WITH ATTRIBUTE aMACID;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
nameBinding(6) macName(1)};

nbMACMonitor NAME BINDING

SUBORDINATE OBJECT CLASS "IEEE802.1F":ewmaMetricMonitor;
NAMED BY SUPERIOR OBJECT CLASS
"ISO/IEC 10165-2":system;
WITH ATTRIBUTE aScannerId;
CREATE WITH-AUTOMATIC-INSTANCE-NAMING;
DELETE ONLY-IF-NO-CONTAINED-OBJECTS;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
nameBinding(6) macMonitor(2)};

### 30A.1.2 DTE MAC entity attributes

**aMACID ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY;
BEHAVIOUR bMACID;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) macID(3)};

**bMACID BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.1;

**aFramesTransmittedOK ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesTransmittedOK;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) framesTransmittedOK(2)};

**bFramesTransmittedOK BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.2;

NOTES

1—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

2—This maps to framesSent (of the mandatory macPackage) in ISO/IEC 10742: 1994.;

**aSingleCollisionFrames ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bSingleCollisionFrames; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) singleCollisionFrames(3)}; |

**bSingleCollisionFrames BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.3;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 103 h.;

**aMultipleCollisionFrames ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bMultipleCollisionFrames; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) multipleCollisionFrames(4)}; |

**bMultipleCollisionFrames BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.4;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 125 h.;

**aFramesReceivedOK ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bFramesReceivedOK; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) framesReceivedOK(5)}; |

**bFramesReceivedOK BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.5;

NOTES

1—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

2—This maps to framesReceived (of the mandatory macPackage) in ISO/IEC 10742: 1994.;

### aFrameCheckSequenceErrors ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bFrameCheckSequenceErrors;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) frameCheckSequenceErrors(6)};

### bFrameCheckSequenceErrors BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.6;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

### aAlignmentErrors ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bAlignmentErrors;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) alignmentErrors(7)};

### bAlignmentErrors BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.7;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

### aOctetsTransmittedOK ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bOctetsTransmittedOK;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) octetsTransmittedOK(8)};

### bOctetsTransmittedOK BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.8;

NOTES

1—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 min.

2—This maps to octetsSent (of the mandatory macPackage) in ISO/IEC 10742: 1994.;

### aFramesWithDeferredXmissions ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesWithDeferredXmissions;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) framesWithDeferredXmissions(9)};

### bFramesWithDeferredXmissions BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.9;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 103 h.;

### aLateCollisions ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bLateCollisions;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) lateCollisions(10)};

### bLateCollisions BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.10;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

### aFramesAbortedDueToXSColls ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesAbortedDueToXSColls;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) framesAbortedDueToXSColls(11)};

### bFramesAbortedDueToXSColls BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.11;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 53 days.;

### aFramesLostDueToIntMACXmitError ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesLostDueToIntMACXmitError;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) framesLostDueToIntMACXmitError(12)};

**bFramesLostDueToIntMACXmitError BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.12;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 16 h.;

**aCarrierSenseErrors ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bCarrierSenseErrors; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) carrierSenseErrors(13)}; |

**bCarrierSenseErrors BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.13;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aOctetsReceivedOK ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bOctetsReceivedOK; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) octetsReceivedOK(14)}; |

**bOctetsReceivedOK BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.14;

NOTES

1—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 min.

2—This maps to octetsReceived (of the mandatory macPackage) in ISO/IEC 10742: 1994.;

**aFramesLostDueToIntMACRcvError ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bFramesLostDueToIntMACRcvError; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) framesLostDueToIntMACRcvError(15)}; |

**bFramesLostDueToIntMACRcvError BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.3.1.1.15;<br>NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.; |

**aPromiscuousStatus ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802Dot3-MgmtAttributeModule.TrueFalse; |
| BEHAVIOUR | bPromiscuousStatus; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) promiscuousStatus(16)}; |

**bPromiscuousStatus BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.3.1.1.16; |

**aReadMulticastAddressList ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802Dot3-MgmtAttributeModule. MulticastAddressList |
| BEHAVIOUR | bReadMulticastAddressList; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) readMulticastAddressList(17)}; |

**bReadMulticastAddressList BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.3.1.1.17; |

**aMulticastFramesXmittedOK ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bMulticastFramesXmittedOK; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) multicastFramesXmittedOK(18)}; |

**bMulticastFramesXmittedOK BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.3.1.1.18;<br>NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.; |

### aBroadcastFramesXmittedOK ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bBroadcastFramesXmittedOK;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) broadcastFramesXmittedOK(19)};

### bBroadcastFramesXmittedOK BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.19;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

### aFramesWithExcessiveDeferral ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bFramesWithExcessiveDeferral;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) framesWithExcessiveDeferral(20)};

### bFramesWithExcessiveDeferral BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.20;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 days.;

### aMulticastFramesReceivedOK ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bMulticastFramesReceivedOK;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) multicastFramesReceivedOK(21)};

### bMulticastFramesReceivedOK BEHAVIOUR

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.21;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

### aBroadcastFramesReceivedOK ATTRIBUTE

DERIVED FROM aCMCounter;
BEHAVIOUR bBroadcastFramesReceivedOK;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) broadcastFramesReceivedOK(22)};

**bBroadcastFramesReceivedOK BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.3.1.1.22;<br>NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.; |

**aInRangeLengthErrors ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bInRangeLengthErrors; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) inRangeLengthErrors(23)}; |

**bInRangeLengthErrors BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.3.1.1.23;<br>NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.; |

**aOutOfRangeLengthField ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bOutOfRangeLengthField; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) outOfRangeLengthField(24)}; |

**bOutOfRangeLengthField BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.3.1.1.24;<br>NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.; |

**aFrameTooLongErrors ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bFrameTooLongErrors; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) frameTooLongErrors(25)}; |

**bFrameTooLongErrors BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.3.1.1.25;<br>NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 61 days.; |

**aMACEnableStatus ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
BEHAVIOUR bMACEnableStatus;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) mACEnableStatus(26)};

**bMACEnableStatus BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.26;

**aTransmitEnableStatus ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
BEHAVIOUR bTransmitEnableStatus;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) transmitEnableStatus(27)};

**bTransmitEnableStatus BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.27;

**aMulticastReceiveStatus ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.TrueFalse;
BEHAVIOUR bMulticastReceiveStatus;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) multicastReceiveStatus(28)};

**bMulticastReceiveStatus BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.28;

**aReadWriteMACAddress ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802CommonDefinitions.MACAddress;
BEHAVIOUR bReadWriteMACAddress;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) modifyMACAddress(29)};

**bReadWriteMACAddress BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.29;

NOTE—This maps to localMACAddress (of the mandatory macPackage) in ISO/IEC 10742: 1994.;

**aCollisionFrames ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AttemptArray;
BEHAVIOUR bCollisionFrames;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) collisionFrames(30)};

**bCollisionFrames BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.1.30;

NOTE—The approximate minimum time for any single counter rollover for 10 Mb/s operation is 103 h.;

### 30A.1.3 DTE MAC entity actions

**acInitializeMAC ACTION**

BEHAVIOUR bInitializeMAC;
MODE CONFIRMED;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) action(9) initializeMAC(1)};

**bInitializeMAC BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.2.1;

**acAddGroupAddress ACTION**

BEHAVIOUR bAddGroupAddress;
MODE CONFIRMED;
WITH INFORMATION SYNTAX IEEE802CommonDefinitions.MACAddress;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) action(9) addGroupAddress(2)};

**bAddGroupAddress BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.2.2;

**acDeleteGroupAddress ACTION**

BEHAVIOUR bDeleteGroupAddress;
MODE CONFIRMED;
WITH INFORMATION SYNTAX IEEE802CommonDefinitions.MACAddress;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) action(9) deleteGroupAddress(3)};

**bDeleteGroupAddress BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.1.2.3;

**acExecuteSelfTest ACTION**

```
        BEHAVIOUR                                bExecuteSelfTestMAC;
        MODE                                     CONFIRMED;
        REGISTERED AS           {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                                action(9) executeSelfTestMAC(4)};
```

**bExecuteSelfTestMAC BEHAVIOUR**

```
        DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.3.1.2.4;
```

## 30A.2 DTE physical entity managed object class

### 30A.2.1 DTE physical entity formal definition

```
oPHYEntity                                       MANAGED OBJECT CLASS

        DERIVED FROM                             "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
                                                 1992":top;
        CHARACTERIZED BY
                pBasic                           PACKAGE
                        ATTRIBUTES               aPHYID                   GET,
                                                 aPHYType                 GET,
                                                 aPHYTypeList             GET,
                                                 aMIIDetect               GET,
                                                 aPHYAdminState           GET;
                ;
        ;
        CONDITIONAL PACKAGES
                pRecommended                     PACKAGE
                        ATTRIBUTES               aSQETestErrors           GET;
                        REGISTERED AS            {iso(1) member-body(2) us(840) 802dot3(10006)
                                                 csmacdmgt(30) package(4)
                                                 phyRecommendedPkg(6)};
                        PRESENT IF               The Recommended Package is implemented.;
                pMultiplePhy                     PACKAGE
                        ACTIONS                  acPHYAdminControl;
                        REGISTERED AS            {iso(1) member-body(2) us(840) 802dot3(10006)
                                                 csmacdmgt(30) package(4)
                                                 phyMultiplePhyPkg(7)};
                        PRESENT IF               There is more than one PHY per MAC.;




                p100MbpsMonitor                  PACKAGE
                        ATTRIBUTES               aSymbolErrorDuringCarrier    GET;
                        REGISTERED AS            {iso(1) member-body(2) us(840) 802dot3(10006)
                                                 csmacdmgt(30) package(4)
```

```
                                        phy100 MbpsMonitor(8)};
                    PRESENT IF          The 100 Mb/s Monitor capability is implemented.;
      REGISTERED AS          {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                             managedObjectClass(3) phyObjectClass(2)};

nbPHYName                               NAME BINDING

      SUBORDINATE OBJECT CLASS          oPHYEntity;
      NAMED BY SUPERIOR OBJECT CLASS
                                        oMACEntity;
      WITH ATTRIBUTE                    aPHYID;
      REGISTERED AS          {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                             nameBinding(6) phyName(3)};


nbPHYMonitor                            NAME BINDING

      SUBORDINATE OBJECT CLASS          "IEEE802.1F":ewmaMetricMonitor;
      NAMED BY SUPERIOR OBJECT CLASS
                                        "ISO/IEC 10165-2":system;
      WITH ATTRIBUTE                    aScannerId;
            CREATE                      WITH-AUTOMATIC-INSTANCE-NAMING;
            DELETE                      ONLY-IF-NO-CONTAINED-OBJECTS;
      REGISTERED AS          {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                             nameBinding(6) phyMonitor(4)};
```

### 30A.2.2 DTE physical entity attributes

**aPHYID ATTRIBUTE**

```
      WITH ATTRIBUTE SYNTAX             IEEE802Dot3-MgmtAttributeModule.OneOfName;
      MATCHES FOR                       EQUALITY;
      BEHAVIOUR                         bPHYID;
      REGISTERED AS          {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                             attribute(7) phyID(31)};
```

**bPHYID BEHAVIOUR**

```
      DEFINED AS             See "BEHAVIOUR DEFINED AS" in 30.3.2.1.1;
```

**aPHYType ATTRIBUTE**

```
      WITH ATTRIBUTE SYNTAX             IEEE802Dot3-
                                        MgmtAttributeModule.PhyTypeValue;
      MATCHES FOR                       EQUALITY;
      BEHAVIOUR                         bPHYType;
      REGISTERED AS          {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                             attribute(7) pHYType(32)};
```

**bPHYType BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.2.1.2;

**aPHYTypeList ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.PhyTypeList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPHYTypeList;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) pHYTypeList(33)};

**bPHYTypeList BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.2.1.3;

**aSQETestErrors ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bSQETestErrors;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) sqeTestErrors(34)};

**bSQETestErrors BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.2.1.4;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aSymbolErrorDuringCarrier ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bSymbolErrorDuringCarrier;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) symbolErrorDuringCarrier(35)};

**bSymbolErrorDuringCarrier BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.3.2.1.5;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aMIIDetect ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.MIIDetect;
MATCHES FOR EQUALITY;
BEHAVIOUR bMIIDetect;

REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) mIIDetect(36)};

**bMIIDetect BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.2.1.6;

**aPHYAdminState ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
PortAdminState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bPHYAdminState;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
attribute(7) pHYAdminState(37)};

**bPHYAdminState BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.2.1.7;

### 30A.2.3 DTE physical entity actions

**acPHYAdminControl ACTION**

BEHAVIOUR bPHYAdminControl;
MODE CONFIRMED;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.
PortAdminState;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
action(9) pHYAdminControl(5)};

**bPHYAdminControl BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.3.2.2.1;

## 30A.3 Repeater managed object class

### 30A.3.1 Repeater, formal definition

```
oRepeater                                MANAGED OBJECT CLASS

     DERIVED FROM                        "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2
                                         1992":top;
     CHARACTERIZED BY
          pRepeaterBasicControl          PACKAGE
               ATTRIBUTES                aRepeaterID                  GET,
                                         aRepeaterType                GET,
                                         aRepeaterGroupCapacity       GET,
                                         aGroupMap                    GET,
                                         aRepeaterHealthState         GET,
                                         aRepeaterHealthText          GET,
                                         aRepeaterHealthData          GET;
               ACTIONS                   acResetRepeater,
                                         acExecuteNonDisruptiveSelfTest;
               NOTIFICATIONS             nRepeaterHealth,
                                         nRepeaterReset,
                                         nGroupMapChange;
          ;
     ;
     CONDITIONAL PACKAGES
          pRepeaterPerfMonitor           PACKAGE
               ATTRIBUTES                aTransmitCollisions          GET;
               REGISTERED AS             {iso(1) member-body(2) us(840) 802dot3(10006)
                                         csmacdmgt(30) package(4)
                                         repeaterPerfMonitorPkg(9)};
               PRESENT IF                The Performance Monitor Capability is implemented.;
     REGISTERED AS             {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                               managedObjectClass(3) repeaterObjectClass(3)};

nbRepeaterName                           NAME BINDING

     SUBORDINATE OBJECT CLASS            repeater;
     NAMED BY SUPERIOR OBJECT CLASS
                                         "ISO/IEC 10165-2":system  AND SUBCLASSES;
     WITH ATTRIBUTE                      aRepeaterID;
     REGISTERED AS             {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                               nameBinding(6) repeaterName(5)};


nbRepeaterMonitor                        NAME BINDING

     SUBORDINATE OBJECT CLASS            "IEEE802.1F":oEWMAMetricMonitor;
     NAMED BY SUPERIOR OBJECT CLASS
                                         "ISO/IEC 10165-2":system  AND SUBCLASSES;
     WITH ATTRIBUTE                      aScannerId;
          CREATE                         WITH-AUTOMATIC-INSTANCE-NAMING;
          DELETE                         ONLY-IF-NO-CONTAINED-OBJECTS;
     REGISTERED AS             {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                               nameBinding(6) repeaterMonitor(6)};
```

### 30A.3.2 Repeater attributes

**aRepeaterID ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY;
BEHAVIOUR bRepeaterID;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) repeaterID(38)};

**bRepeaterID BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.1.1.1;

**aRepeaterType ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.RepeaterType;
MATCHES FOR EQUALITY;
BEHAVIOUR bRepeaterType;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) repeaterType (39)};

**bRepeaterType BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.1.1.2;

**aRepeaterGroupCapacity ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bRepeaterGroupCapacity;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) repeaterGroupCapacity(40)};

**bRepeaterGroupCapacity BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.1.1.3;

**aGroupMap ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.BitString;
MATCHES FOR EQUALITY;
BEHAVIOUR bGroupMap;

REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) groupMap(41)};

**bGroupMap BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.1.1.4;

**aRepeaterHealthState ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802Dot3-MgmtAttributeModule.<br>RepeaterHealthState; |
| MATCHES FOR | EQUALITY; |
| BEHAVIOUR | bRepeaterHealthState; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) repeaterHealthState(42)}; |

**bRepeaterHealthState BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.4.1.1.5; |

**aRepeaterHealthText ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802Dot3-MgmtAttributeModule.<br>RepeaterHealthText; |
| MATCHES FOR | EQUALITY; |
| BEHAVIOUR | bRepeaterHealthText; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) repeaterHealthText(43)}; |

**bRepeaterHealthText BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.4.1.1.6; |

**aRepeaterHealthData ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802Dot3-MgmtAttributeModule.<br>RepeaterHealthData; |
| MATCHES FOR | EQUALITY; |
| BEHAVIOUR | bRepeaterHealthData; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) repeaterHealthData(44)}; |

**bRepeaterHealthData BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See "BEHAVIOUR DEFINED AS" in 30.4.1.1.7; |

**aTransmitCollisions ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bTransmitCollisions; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) transmitCollisions (45)}; |

**bTransmitCollisions BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.1.1.8;

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.;

### 30A.3.3 Repeater actions

**acResetRepeater ACTION**

BEHAVIOUR bResetRepeater;
MODE CONFIRMED;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) action(9) resetRepeater(6)};

**bResetRepeater BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.1.2.1;

**acExecuteNonDisruptiveSelfTest ACTION**

BEHAVIOUR bExecuteNonDisruptiveSelfTest;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) action(9) executeNonDisruptiveSelfTestAction(7)};

**bExecuteNonDisruptiveSelfTest BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.1.2.2;

### 30A.3.4 Repeater notifications

**nRepeaterHealth NOTIFICATION**

BEHAVIOUR bRepeaterHealth;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.RepeaterHealthInfo
AND ATTRIBUTE IDS repeaterHealthState aRepeaterHealthState,
repeaterHealthText aRepeaterHealthText,
repeaterHealthData aRepeaterHealthData
;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) notification(10) repeaterHealth(1)};

**bRepeaterHealth BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.1.3.1;

**nRepeaterReset NOTIFICATION**

```
        BEHAVIOUR                               bRepeaterReset;
        WITH INFORMATION SYNTAX                 IEEE802Dot3-MgmtAttributeModule.
                                                RepeaterHealthInfo
        AND ATTRIBUTE IDS                       repeaterHealthState aRepeaterHealthState,
                                                repeaterHealthText aRepeaterHealthText,
                                                repeaterHealthData aRepeaterHealthData
        ;
        REGISTERED AS           {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                                notification(10) repeaterReset(2)};
```

**bRepeaterReset BEHAVIOUR**

```
        DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.4.1.3.2;
```

**nGroupMapChange NOTIFICATION**

```
        BEHAVIOUR                               bGroupMapChange;
        WITH INFORMATION SYNTAX                 IEEE802Dot3-MgmtAttributeModule.BitString;
        REGISTERED AS           {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                                notification(10) groupMapChange(3)};
```

**bGroupMapChange BEHAVIOUR**

```
        DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.4.1.3.3;
```

## 30A.4 Group managed object class

### 30A.4.1 Group, formal definition

```
oGroup                                          MANAGED OBJECT CLASS

        DERIVED FROM                            "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2
                                                1992":top;
        CHARACTERIZED BY
                pGroupBasicControl              PACKAGE
                        ATTRIBUTES              aGroupID                        GET,
                                                aGroupPortCapacity              GET,
                                                aPortMap                        GET;
                        NOTIFICATIONS           nPortMapChange;
                ;
        ;

        REGISTERED AS           {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                                managedObjectClass(3) groupObjectClass(4)};

nbGroupName                                     NAME BINDING

        SUBORDINATE OBJECT CLASS                oGroup;
        NAMED BY SUPERIOR OBJECT CLASS
```

```
                                         oRepeater  AND SUBCLASSES;
     WITH ATTRIBUTE                      aGroupID;
     REGISTERED AS             {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                               nameBinding(6) groupName(7)};
```

### 30A.4.2 Group attributes

**aGroupID ATTRIBUTE**

```
     WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.OneOfName;
     MATCHES FOR                         EQUALITY;
     BEHAVIOUR                           bGroupID;
     REGISTERED AS             {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                               attribute(7) groupID(46)};
```

**bGroupID BEHAVIOUR**

```
     DEFINED AS                See "BEHAVIOUR DEFINED AS" in 30.4.2.1.1;
```

**aGroupPortCapacity ATTRIBUTE**

```
     WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.OneOfName;
     MATCHES FOR                         EQUALITY, ORDERING;
     BEHAVIOUR                           bGroupPortCapacity;
     REGISTERED AS             {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                               attribute(7) groupPortCapacity(47)};
```

**bGroupPortCapacity BEHAVIOUR**

```
     DEFINED AS                See "BEHAVIOUR DEFINED AS" in 30.4.2.1.2;
```

**aPortMap ATTRIBUTE**

```
     WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.BitString;
     MATCHES FOR                         EQUALITY;
     BEHAVIOUR                           bPortMap;
     REGISTERED AS             {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                               attribute(7) portMap(48)};
```

**bPortMap BEHAVIOUR**

```
     DEFINED AS                See "BEHAVIOUR DEFINED AS" in 30.4.2.1.3;
```

### 30A.4.3 Group notifications

**nPortMapChange NOTIFICATION**

```
     BEHAVIOUR                           bPortMapChange;
     WITH INFORMATION SYNTAX             IEEE802Dot3-MgmtAttributeModule.BitString;
```

```
        REGISTERED AS          {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                               notification(10) portMapChange(4)};
```

**bPortMapChange BEHAVIOUR**

```
        DEFINED AS             See “BEHAVIOUR DEFINED AS” in 30.4.2.2.1;
```

## 30A.5 Repeater port managed object class

### 30A.5.1 Port, formal definition

```
oRepeaterPort                          MANAGED OBJECT CLASS

        DERIVED FROM                   “CCITT Rec. X.721 (1992) | ISO/IEC 10165-2
                                       1992”:top;
        CHARACTERIZED BY
                pPortBasicControl      PACKAGE
                        ATTRIBUTES     aPortID                        GET,
                                       aPortAdminState                GET,
                                       aAutoPartitionState            GET;
                        ACTIONS        acPortAdminControl;
                ;
        ;
        CONDITIONAL PACKAGES
                pPortPerfMonitor       PACKAGE
                        ATTRIBUTES     aReadableFrames                GET,
                                       aReadableOctets                GET,
                                       aFrameCheckSequenceErrors      GET,
                                       aAlignmentErrors               GET,
                                       aFramesTooLong                 GET,
                                       aShortEvents                   GET,
                                       aRunts                         GET,
                                       aCollisions                    GET,
                                       aLateEvents                    GET,
                                       aVeryLongEvents                GET,
                                       aDataRateMismatches            GET,
                                       aAutoPartitions                GET;
                        REGISTERED AS  {iso(1) member-body(2) us(840) 802dot3(10006)
                                       csmacdmgt(30) package(4)
                                       portPerfMonitorPkg(10)};
                        PRESENT IF     The Performance Monitor Capability is implemented.;
                pPortAddrTracking      PACKAGE
                        ATTRIBUTES     aLastSourceAddress             GET,
                                       aSourceAddressChanges          GET;
                        REGISTERED AS  {iso(1) member-body(2) us(840) 802dot3(10006)
                                       csmacdmgt(30) package(4) portAddrTrackPkg(11)};
                        PRESENT IF     The Address Tracking and Performance Monitor
                                       capabilities are implemented.;
                p100MbpsMonitor        PACKAGE
                        ATTRIBUTES     aIsolates                      GET,
                                       aSymbolErrorDuringPacket       GET;
                        REGISTERED AS  {iso(1) member-body(2) us(840) 802dot3(10006)
```

```
                                        csmacdmgt(30) package(4)
                                        port100 MbpsMonitor(12)};
                        PRESENT IF      The 100 Mb/s Monitor capability is implemented;
    REGISTERED AS           {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                            managedObjectClass(3) repeaterPortObjectClass(5)};

nbPortName                              NAME BINDING

    SUBORDINATE OBJECT CLASS            oRepeaterPort;
    NAMED BY SUPERIOR OBJECT CLASS
                                        oGroup  AND SUBCLASSES;
    WITH ATTRIBUTE                      aPortID;
    REGISTERED AS           {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                            nameBinding(6) portName(8)};
```

### 30A.5.2 Port attributes

**aPortID ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.OneOfName;
    BEHAVIOUR                           bPortID;
    REGISTERED AS           {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                            attribute(7) portID(49)};
```

**bPortID BEHAVIOUR**

```
    DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.4.3.1.1;
```

**aPortAdminState ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.
                                        PortAdminState;
    MATCHES FOR                         EQUALITY;
    BEHAVIOUR                           bPortAdminState;
    REGISTERED AS           {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                            attribute(7) portAdminState(50)};
```

**bPortAdminState BEHAVIOUR**

```
    DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.4.3.1.2;
```

**aAutoPartitionState ATTRIBUTE**

```
    WITH ATTRIBUTE SYNTAX               IEEE802Dot3-MgmtAttributeModule.
                                        AutoPartitionState;
    MATCHES FOR                         EQUALITY;
    BEHAVIOUR                           bAutoPartition;
    REGISTERED AS           {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                            attribute(7) autoPartitionState(51)};
```

**bAutoPartition BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.3;

**aReadableFrames ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bReadableFrames;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) readableFrames(52)};

**bReadableFrames BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.4;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aReadableOctets ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bReadableOctets;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) readableOctets(53)};

**bReadableOctets BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.5;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 58 min.;

**aFrameCheckSequenceErrors ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bFCSErrors;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) frameCheckSequenceErrors(54)};

**bFCSErrors BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.6;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aAlignmentErrors ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bAlignmentErrors; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) alignmentErrors(55)}; |

**bAlignmentErrors BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.3.1.7;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 80 h.;

**aFramesTooLong ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bFramesTooLong; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) framesTooLong(56)}; |

**bFramesTooLong BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.3.1.8;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 61 days.;

**aShortEvents ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bShortEvents; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) shortEvents(57)}; |

**bShortEvents BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.3.1.9;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 16 hours;

**aRunts ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bRunts; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) runts(58)}; |

**bRunts BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.10;

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.;

**aCollisions ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bCollisions;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) collisions(59)};

**bCollisions BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.11;

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 16 h.;

**aLateEvents ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bLateEvents;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) lateEvents(60)};

**bLateEvents BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.12;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 81 h.;

**aVeryLongEvents ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bVeryLongEvents;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) veryLongEvents(61)};

**bVeryLongEvents BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.13;

NOTE—The approximate minimum time between counter rollovers for 10 Mb/s operation is 198 days.;

**aDataRateMismatches ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bDataRateMismatches; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) dataRateMismatches(62)}; |

**bDataRateMismatches BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.14;

**aAutoPartitions ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bAutoPartitions; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) autoPartitions(63)}; |

**bAutoPartitions BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.15;

**aIsolates ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bIsolates; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) isolates(64)}; |

**bIsolates BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.16;

**aSymbolErrorDuringPacket ATTRIBUTE**

| | |
|---|---|
| DERIVED FROM | aCMCounter; |
| BEHAVIOUR | bSymbolErrorDuringPacket; |
| REGISTERED AS | {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) symbolErrorDuringPacket(65)}; |

**bSymbolErrorDuringPacket BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.4.3.1.17;

**aLastSourceAddress ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802CommonDefinitions.MACAddress; |
| MATCHES FOR | EQUALITY; |

BEHAVIOUR bLastSourceAddress;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) lastSourceAddress(66)};

**bLastSourceAddress BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.3.1.18;

**aSourceAddressChanges ATTRIBUTE**

DERIVED FROM aCMCounter;
BEHAVIOUR bSourceAddressChanges;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) sourceAddressChanges(67)};

**bSourceAddressChanges BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.3.1.19;

NOTE—The approximate minimum time for counter rollover for 10 Mb/s operation is 81 h.;

### 30A.5.3 Port actions

**acPortAdminControl ACTION**

BEHAVIOUR bPortAdminControl;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.PortAdminState;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) action(9) portAdminControl(8)};

**bPortAdminControl BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.4.3.2.1;

## 30A.6 MAU managed object class

### 30A.6.1 MAU, formal definition

oMAU MANAGED OBJECT CLASS

DERIVED FROM “CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 : 1992”:top;

CHARACTERIZED BY
pMAUBasic PACKAGE
ATTRIBUTES aMAUID GET,
aMAUType GET-SET,

```
                                    aMAUTypeList                GET,
                                    aMediaAvailable             GET,
                                    aJabber                     GET,
                                    aMAUAdminState              GET;
                  NOTIFICATIONS     nJabber;
            ;
      ;
      CONDITIONAL PACKAGES
            pMAUControl                 PACKAGE
                  ACTIONS               acResetMAU,
                                        acMAUAdminControl;
                  REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4) mauControlPkg(13)};
                  PRESENT IF            The pMAUControl package is implemented.;

            pMediaLossTracking          PACKAGE
                  ATTRIBUTES            aLoseMediaCounter           GET;
                  REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4) mediaLossTrackingPkg(14)};
                  PRESENT IF            MAU TypeValue = AUI or if the
                                        pMediaLossTracking package is implemented.;

            pBroadbandDTEMAU            PACKAGE
                  ATTRIBUTES            aBbMAUXmitRcvSplitType      GET,
                                        aBroadbandFrequencies       GET;
                  REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4) broadbandMAUPkg(15)};
                  PRESENT IF            The MAU is of type 10BROAD36.;

            p100MbpsMonitor             PACKAGE
                  ATTRIBUTES            aFalseCarriers              GET;
                  REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        package(4) mau100MbpsMonitor(16)};
                  PRESENT IF            The MAU is capable of 100 Mb/s operation.;
      REGISTERED AS                     {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
                                        managedObjectClass(3) mauObjectClass(6)};
nbMAU-repeaterName                      NAME BINDING

      SUBORDINATE OBJECT CLASS          oMAU;
      NAMED BY SUPERIOR OBJECT CLASS                 --(of oRepeaterPort)
                                        oRepeaterPort  AND SUBCLASSES;
                                                     --{1.2.840.10006.30.3.5}
      WITH ATTRIBUTE                    aMAUID;
      REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                              mau-repeaterName(9)};

nbMAU-dteName                           NAME BINDING

      SUBORDINATE OBJECT CLASS          oMAU;
      NAMED BY SUPERIOR OBJECT CLASS                 --(of oPHYEntity)
                                        oPHYEntity  AND SUBCLASSES
                                                     --{1.2.840.10006.30.3.2};
      WITH ATTRIBUTE                    aMAUID;
      REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
```

```
                        mau-dteName(10)};
```

## 30A.6.2 MAU attributes

**aMAUID ATTRIBUTE**

```
WITH ATTRIBUTE SYNTAX             IEEE802Dot3-MgmtAttributeModule.OneOfName;
MATCHES FOR                       EQUALITY;
BEHAVIOUR                         bMAUID;
REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                        mauID(68)};
```

**bMAUID BEHAVIOUR**

```
DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.5.1.1.1;
```

**aMAUType ATTRIBUTE**

```
WITH ATTRIBUTE SYNTAX             IEEE802Dot3-MgmtAttributeModule.TypeValue;
MATCHES FOR                       EQUALITY, ORDERING;
BEHAVIOUR                         bMAUType;
REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                        mauType(69)};
```

**bMAUType BEHAVIOUR**

```
DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.5.1.1.2;
```

**aMAUTypeList ATTRIBUTE**

```
WITH ATTRIBUTE SYNTAX             IEEE802Dot3-MgmtAttributeModule.TypeList;
MATCHES FOR                       EQUALITY, ORDERING;
BEHAVIOUR                         bMAUTypeList;
REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                        mauTypeList(70)};
```

**bMAUTypeList BEHAVIOUR**

```
DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.5.1.1.3;
```

**aMediaAvailable ATTRIBUTE**

```
WITH ATTRIBUTE SYNTAX             IEEE802Dot3-MgmtAttributeModule.
                                  MediaAvailState;
MATCHES FOR                       EQUALITY, ORDERING;
BEHAVIOUR                         bMediaAvailable;
REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                        mauMediaAvailable(71)};
```

**bMediaAvailable BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.1.4;

**aLoseMediaCounter ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.aCMCounter;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bLoseMediaCounter;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauLoseMediaCounter(72)};

**bLoseMediaCounter BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.1.5;

**aJabber ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.Jabber;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bJabberAttribute;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) jabber(73)};

**bJabberAttribute BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.1.6;

**aMAUAdminState ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.AdminState;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bMAUAdminState;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) mauAdminState(74)};

**bMAUAdminState BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.5.1.1.7;

**aBbMAUXmitRcvSplitType ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.BbandXmitRcvSplitType;
MATCHES FOR EQUALITY;
BEHAVIOUR bBbMAUXmitRcvSplitType;

REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) bBandSplitType(75)};

**bBbMAUXmitRcvSplitType BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.8;

**aBroadbandFrequencies ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802Dot3-MgmtAttributeModule.<br>BbandFrequency; |
| MATCHES FOR | EQUALITY; |
| BEHAVIOUR | bBroadbandFrequencies; |
| REGISTERED AS | {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) bBandFrequencies(76)}; |

**bBroadbandFrequencies BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.9;

**aFalseCarriers ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802Dot3-MgmtAttributeModule.aCMCounter; |
| MATCHES FOR | EQUALITY, ORDERING; |
| BEHAVIOUR | bFalseCarriers; |
| REGISTERED AS | {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) falseCarriers(77)}; |

**bFalseCarriers BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.1.10;

### 30A.6.3 MAU actions

**acResetMAU ACTION**

| | |
|---|---|
| BEHAVIOUR | bResetMAU; |
| MODE | CONFIRMED; |
| REGISTERED AS | {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9) resetMAU(9)}; |

**bResetMAU BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.5.1.2.1;

**acMAUAdminControl ACTION**

```
        BEHAVIOUR                               bMAUAdminControl;
        WITH INFORMATION SYNTAX                 IEEE802Dot3-MgmtAttributeModule.AdminState;
        MODE                                    CONFIRMED;
        REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)
                                mauAdminCtrl(10)};
```

**bMAUAdminControl BEHAVIOUR**

```
        DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.5.1.2.2;
```

### 30A.6.4 MAU notifications

**nJabber NOTIFICATION**

```
        BEHAVIOUR                               bJabberNotification;
        WITH INFORMATION SYNTAX                 IEEE802Dot3-MgmtAttributeModule.Jabber;
        ;
        REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) notification(10)
                                jabber(5)};
```

**bJabberNotification BEHAVIOUR**

```
        DEFINED AS              See "BEHAVIOUR DEFINED AS" in 30.5.1.3.1;
```

## 30A.7 AutoNegotiation managed object class

### 30A.7.1 AutoNegotiation, formal definition

```
oAutoNegotiation                                MANAGED OBJECT CLASS

        DERIVED FROM                            "CCITT Rec. X.721 (1992) | ISO/IEC 10165-2 :
                                                1992":top;
        CHARACTERIZED BY
                pAutoNeg                        PACKAGE
                        ATTRIBUTES              aAutoNegID                              GET,
                                                aAutoNegAdminState                      GET,
                                                aAutoNegRemoteSignaling                 GET,
                                                aAutoNegAutoConfig                      GET-SET,
                                                aAutoNegLocalTechnologyAbility          GET,
                                                aAutoNegAdvertisedTechnologyAbility     GET-SET,
                                                aAutoNegReceivedTechnologyAbility       GET,
                                                aAutoNegLocalSelectorAbility            GET,
                                                aAutoNegAdvertisedSelectorAbility       GET-SET,
                                                aAutoNegReceivedSelectorAbility         GET;
                        ACTIONS
                                                acAutoNegRestartAutoConfig,
                                                acAutoNegAdminControl;
                ;
        ;
        REGISTERED AS           {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30)
```

```
                              managedObjectClass(3) autoNegObjectClass(7)};

nbAutoNeg-mauName                         NAME BINDING

        SUBORDINATE OBJECT CLASS          oMAU;
        NAMED BY SUPERIOR OBJECT CLASS                    --(of oMAU)
                                          oMAU   AND SUBCLASSES;
                                                          --{1.2.840.10006.30.3.6}
        WITH ATTRIBUTE                    aMAUID;
        REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) nameBinding(6)
                              autoNeg-mauName(11)};
```

### 30A.7.2 Auto-Negotiation attributes

**aAutoNegID ATTRIBUTE**

```
        WITH ATTRIBUTE SYNTAX             IEEE802Dot3-MgmtAttributeModule.OneOfName;
        MATCHES FOR                       EQUALITY;
        BEHAVIOUR                         bAutoNegID;
        REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                              autoNegID(78)};
```

**bAutoNegID BEHAVIOUR**

```
        DEFINED AS            See "BEHAVIOUR DEFINED AS" in 30.6.1.1.1;
```

**aAutoNegAdminState ATTRIBUTE**

```
        WITH ATTRIBUTE SYNTAX             IEEE802Dot3-MgmtAttributeModule.
                                          AutoNegAdminState;
        MATCHES FOR                       EQUALITY;
        BEHAVIOUR                         bAutoNegAdminState;
        REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                              autoNegAdminState(79)};
```

**bAutoNegAdminState BEHAVIOUR**

```
        DEFINED AS            See "BEHAVIOUR DEFINED AS" in 30.6.1.1.2;
```

**aAutoNegRemoteSignaling ATTRIBUTE**

```
        WITH ATTRIBUTE SYNTAX             IEEE802Dot3-MgmtAttributeModule.
                                          AutoNegRemoteSignalingDetect;
        MATCHES FOR                       EQUALITY;
        BEHAVIOUR                         bAutoNegRemoteSignaling;
        REGISTERED AS         {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
                              autoNegRemoteSignaling(80)};
```

**bAutoNegRemoteSignaling BEHAVIOUR**

```
        DEFINED AS            See "BEHAVIOUR DEFINED AS" in 30.6.1.1.3;
```

**aAutoNegAutoConfig ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegAutoConfig;
MATCHES FOR EQUALITY;
BEHAVIOUR bAutoNegAutoConfig;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
autoNegAutoConfig(81)};

**bAutoNegAutoConfig BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.6.1.1.4;

**aAutoNegLocalTechnologyAbility ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegTechnologyList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bAutoNegLocalTechnologyAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
autoNegLocalTechnologyAbility(82)};

**bAutoNegLocalTechnologyAbility BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.6.1.1.5;

**aAutoNegAdvertisedTechnologyAbility ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegTechnologyList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bAutoNegAdvertisedTechnologyAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)
autoNegAdvertisedTechnologyAbility(83)};

**bAutoNegAdvertisedTechnologyAbility BEHAVIOUR**

DEFINED AS See “BEHAVIOUR DEFINED AS” in 30.6.1.1.6;

**aAutoNegReceivedTechnologyAbility ATTRIBUTE**

WITH ATTRIBUTE SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegTechnologyList;
MATCHES FOR EQUALITY, ORDERING;
BEHAVIOUR bAutoNegReceivedTechnologyAbility;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7)

autoNegReceivedTechnologyAbility(84)};

**bAutoNegReceivedTechnologyAbility BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See “BEHAVIOUR DEFINED AS” in 30.6.1.1.7; |

**aAutoNegLocalSelectorAbility ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802Dot3-MgmtAttributeModule.AutoNegSelectorList; |
| MATCHES FOR | EQUALITY, ORDERING; |
| BEHAVIOUR | bAutoNegLocalSelectorAbility; |
| REGISTERED AS | {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegLocalSelectorAbility(85)}; |

**bAutoNegLocalSelectorAbility BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See “BEHAVIOUR DEFINED AS” in 30.6.1.1.8; |

**aAutoNegAdvertisedSelectorAbility ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802Dot3-MgmtAttributeModule.AutoNegSelectorList; |
| MATCHES FOR | EQUALITY, ORDERING; |
| BEHAVIOUR | bAutoNegAdvertisedSelectorAbility; |
| REGISTERED AS | {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegAdvertisedSelectorAbility(86)}; |

**bAutoNegAdvertisedSelectorAbility BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See “BEHAVIOUR DEFINED AS” in 30.6.1.1.9; |

**aAutoNegReceivedSelectorAbility ATTRIBUTE**

| | |
|---|---|
| WITH ATTRIBUTE SYNTAX | IEEE802Dot3-MgmtAttributeModule.AutoNegSelectorList; |
| MATCHES FOR | EQUALITY, ORDERING; |
| BEHAVIOUR | bAutoNegReceivedSelectorAbility; |
| REGISTERED AS | {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) attribute(7) autoNegReceivedSelectorAbility(87)}; |

**bAutoNegReceivedSelectorAbility BEHAVIOUR**

| | |
|---|---|
| DEFINED AS | See “BEHAVIOUR DEFINED AS” in 30.6.1.1.10; |

### 30A.7.3 AutoNegotiation actions

**acAutoNegRestartAutoConfig ACTION**

BEHAVIOUR bAutoNegRestartAutoConfig;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)
autoNegRestartAutoConfig(11)};

**bAutoNegRestartAutoConfig BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.2.1;

**acAutoNegAdminControl ACTION**

BEHAVIOUR bAutoNegAdminControl;
WITH INFORMATION SYNTAX IEEE802Dot3-MgmtAttributeModule.
AutoNegAdminState;
MODE CONFIRMED;
REGISTERED AS {iso(1) std(0) iso8802(8802) csma(3) csmacdmgt(30) action(9)
autoNegAdminCtrl(12)};

**bAutoNegAdminControl BEHAVIOUR**

DEFINED AS See "BEHAVIOUR DEFINED AS" in 30.6.1.2.2;

## 30A.8 ResourceTypeID managed object class

### 30A.8.1 ResourceTypeID, formal definition

— Implementation of this managed object in accordance with the definition contained in IEEE Std
— 802.1F-1993 is a conformance requirement of this standard.
— NOTE—A single instance of the Resource Type ID managed object exists within the oMACEntity
— managed object class, a single instance of the Resource Type ID managed object exists within
— the oRepeater managed object class, and a single instance of the Resource Type ID managed
— object exists within the oMAU managed object class conditional on the presence of an MII.
— The managed object itself is contained in IEEE Std 802.1F-1993, therefore only name bindings
— appear in this standard;

nbResourceTypeID-mac NAME BINDING

SUBORDINATE OBJECT CLASS "IEEE802.1F":oResourceTypeID;
NAMED BY SUPERIOR OBJECT CLASS
oMACEntity;
WITH ATTRIBUTE "IEEE802.1F":aResourceTypeIDName;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
nameBinding(6) resourceTypeID-mac(12)};

nbResourceTypeID-repeater NAME BINDING

```
        SUBORDINATE OBJECT CLASS          "IEEE802.1F":oResourceTypeID;
        NAMED BY SUPERIOR OBJECT CLASS
                                          oRepeater  AND SUBCLASSES;
        WITH ATTRIBUTE                    "IEEE802.1F":aResourceTypeIDName;
        REGISTERED AS         {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                              nameBinding(6) resourceTypeID-repeater(13)};

nbResourceTypeID-mau                      NAME BINDING

        SUBORDINATE OBJECT CLASS          "IEEE802.1F":oResourceTypeID;
        NAMED BY SUPERIOR OBJECT CLASS
                                          oMAU  AND SUBCLASSES;
        WITH ATTRIBUTE                    "IEEE802.1F":aResourceTypeIDName;
        REGISTERED AS         {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30)
                              nameBinding(6) resourceTypeID-mau(14)};
```

# Annex 30B

(normative)

# GDMO and ASN.1 definitions for management

## 30B.1 Common attributes template

**aCMCounter ATTRIBUTE**

DERIVED FROM "ISO/IEC 10165-5":genericWrappingCounter;
BEHAVIOUR bCMCounter;
REGISTERED AS {iso(1) member-body(2) us(840) 802dot3(10006) csmacdmgt(30) attribute(7) cmCounter(88)};

**bCMCounter BEHAVIOUR**

DEFINED AS Wraps at one of two sizes. Size is conditional.

Wraps at 32 bits, that is this counter reaches its maximum value at $2^{32}-1$ (i.e., approximately $4.294 \times 10^9$) and then rolls over to zero on the next increment, if maximum increment rate from zero causes a rollover in 58 min or more.

Wraps at 64 bits, that is this counter reaches its maximum value at $2^{64}-1$ (i.e., approximately $1.844 \ldots \times 10^{19}$) and then rolls over to zero on the next increment, if maximum increment rate from zero would cause a 32 bit counter to roll over in less than 58 min.

The counter that this is derived from initializes to zero. Initialization to zero is not a requirement of this standard;

## 30B.2 ASN.1 module for CSMA/CD managed objects

This ASN.1 module defines the ASN.1 types and subtypes that are referred to immediately after the WITH ATTRIBUTE SYNTAX construct in this clause's uses of the attribute template defined in ISO/IEC 10165-4: 1992, Guidelines for the definition of managed objects (GDMO).

IEEE802Dot3-MgmtAttributeModule {iso(1) member-body(2) us(840) 802dot3(10006) global(1) asn1Module(2) commonDefinitions(0) version(2)} DEFINITIONS IMPLICIT TAGS::= BEGIN

EXPORTS--*everything*

IMPORTS--*implicitly imports ISO 8824: 1990*

```
    MACAddress
        FROM IEEE802CommonDefinitions
        {iso(1) member-body(2) us(840) ieee802dot1partF(10011)
        asn1Module(2) commonDefinitions(0) version1(0)};


    AdminState::= ENUMERATED {
        other                   (1),        --undefined
```

```
          unknown                 (2),      --initializing, true state not yet known
          operational             (3),      --powered and connected
          standby                 (4),      --inactive but on
          shutdown                (5)       --similar to power down
          }


     AttemptArray::= SEQUENCE OF aCMCounter--array [1..attempt limit - 1]


     AutoNegAdminState::= ENUMERATED {
          disabled                (1),
          enabled                 (2)
          }


     AutoNegAutoConfig:=ENUMERATED {
          other                   (1),
          configuring             (2),
          complete                (3),
          disabled                (4),
          parallel detect fail    (5)
          }


     AutoNegRemoteSignalingDetect::=ENUMERATED {
          detected                (1),
          notdetected             (2)
          }


     AutoNegSelector::=ENUMERATED {
          other                   (1),      --undefined
          ethernet                (2),      --802.3
          isoethernet             (3)       --802.9
          }


     AutoNegSelectorList::=SEQUENCE OF AutoNegSelector


     AutoNegTechnology::=ENUMERATED {
          global                  (0),      --reserved for future use.
          other                   (1),      --undefined
          unknown                 (2),      --initializing, true ability not yet known.
          10BASE-T                (14),     --10BASE-T as defined in clause 14
          100BASE-T4              (23),     --100BASE-T4 as defined in clause 23
          100BASE-TX              (25),     --100BASE-TX as defined in clause 25
          10BASE-TFD              (142),    --Full-duplex 10BASE-T
          100BASE-TXFD            (252),    --Full-duplex 100BASE-TX
          isoethernet             (8029)    --802.9 ISLAN-16T
          }
     AutoNegTechnologyList::=SEQUENCE OF AutoNegTechnology

     AutoPartitionState::= ENUMERATED {
          autoPartitioned         (1),
          notAutoPartitioned      (2)
          }


     BbandFrequency::= SEQUENCE {
          xmitCarrierFrequency    [1]       INTEGER , --Frequency in MHz times 4 (250 kHz resolution)
          translationFrequency    [2]       INTEGER   --Frequency in MHz times 4 (250 kHz resolution)
          }
```

```
BbandXmitRcvSplitType::= ENUMERATED {
        other                   (1),      --undefined
        single                  (2),      --single-cable system
        dual                    (3)       --dual-cable system, offset normally zero
        }


BitString ::= BIT STRING (SIZE (1..1024))


Jabber::= SEQUENCE {
        jabberFlag              [1]       JabberFlag,
        jabberCounter           [2]       JabberCounter
        }


JabberFlag::= ENUMERATED {
        other                   (1),      --undefined
        unknown                 (2),      --initializing, true state not yet known
        normal                  (3),      --state is true or normal
        fault                   (4)       --state is false, fault or abnormal
        }
```

JabberCounter::= INTEGER (0..$2^{32}-1$)

```
MauTypeList::= SEQUENCE OF TypeValue


MediaAvailState::= ENUMERATED {
        other                   (1),      --undefined
        unknown                 (2),      --initializing, true state not yet known
        available               (3),      --link or light normal, loopback normal
        not available           (4),      --link loss or low light, no loopback
        remote fault            (5),      --remote fault with no detail
        invalid signal          (6),      --invalid signal, applies only to 10BASE-FB
        remote jabber           (7),      --remote fault, reason known to be jabber
        remote link loss        (8),      --remote fault, reason known to be far-end link loss
        remote test             (9)       --remote fault, reason known to be test
        }


MIIDetect::= ENUMERATED {
        unknown                    (1),
        presentNothingConnected    (2),
        presentConnected           (3),
        absent                     (4)
        }


MulticastAddressList::= SEQUENCE OF MACAddress


OneOfName ::= INTEGER (1..1024)


PhyTypeList::= SEQUENCE OF PhyTypeValue


PhyTypeValue::= ENUMERATED {
        other                   (1),      --undefined:
        unknown                 (2),      --initializing, true state or type not yet known
```

```
          none                 (3),      --MII present and nothing connected
          10 Mb/s              (7),      --clause 7 10 Mb/s Manchester
          100BASE-T4           (23),     --clause 23 100 Mb/s 8B/6T
          100BASE-X            (24)      --clause 24 100 Mb/s 4B/5B
          }


     PortAdminState::= ENUMERATED {
          disabled             (1),
          enabled              (2)
          }


     RepeaterHealthData ::= OCTET STRING (SIZE (0..255))


     RepeaterHealthInfo::= SEQUENCE {
          repeaterHealthState    [1]     RepeaterHealthState,
          repeaterHealthText     [2]     RepeaterHealthText  OPTIONAL,
          repeaterHealthData     [3]     RepeaterHealthData  OPTIONAL
          }


     RepeaterHealthState::= ENUMERATED {
          other                (1),      --undefined or unknown
          ok                   (2),      --no known failures
          repeaterFailure      (3),      --known to have a repeater-related failure
          groupFailure         (4),      --known to have a group-related failure
          portFailure          (5),      --known to have a port-related failure
          generalFailure       (6)       --has a failure condition, unspecified type
          }


     RepeaterType::= ENUMERATED {
          other                (1),      --See 20.2.2.3:
          unknown              (2),      --initializing, true state or type not yet known
          10 Mb/s              (9),      --clause 9 10 Mb/s Baseband repeater
          100 Mb/sClassI       (271),    --clause 27 class I 100 Mb/s Baseband repeater
          100 Mb/sClassII      (272),    --clause 27 class II 100 Mb/s Baseband repeater
          802.9a               (99)      --Integrated services repeater
          }


     RepeaterHealthText ::= PrintableString (SIZE (0..255))


     TrueFalse ::= BOOLEAN


     TypeList::= SEQUENCE OF TypeValue


     TypeValue::= ENUMERATED {
          global               (0),      --undefined
          other                (1),      --undefined
          unknown              (2),      --initializing, true state not yet known
          AUI                  (7),      --no internal MAU, view from AUI
          10BASE5              (8),      --Thick coax MAU as specified in clause 8
          FOIRL                (9),      --FOIRL MAU as specified in 9.9
          10BAS                (10),     --Thin coax MAU as specified in clause 10
          10BROAD36            (11),     --Broadband DTE MAU as specified in clause 11
          10BASE-T             (14),     --UTP MAU as specified in clause 14
          10BASE-FP            (16),     --Passive fiber MAU, specified in clause 16
          10BASE-FB            (17),     --Synchronous fiber MAU, specified in clause 17
```

```
        10BASE-FL          (18),    --Asynchronous fiber  MAU, specified in clause 18
        100BASE-T4         (23),    --Four-pair Category 3 UTP as specified in clause 23
        100BASE-TX         (25),    --Two-pair Category 5 UTP as specified in clause 25
        100BASE-FX         (26),    --X fiber over PMD as specified in clause 26
        802.9a             (99)     --Integrated services MAU as specified in IEEE Std 802.9
                                      ISLAN-16T
        }

    END
```