The PCS shall implement the Carrier Sense process as depicted in figure 24-12 including compliance with the associated state variables as specified in 24.2.3.

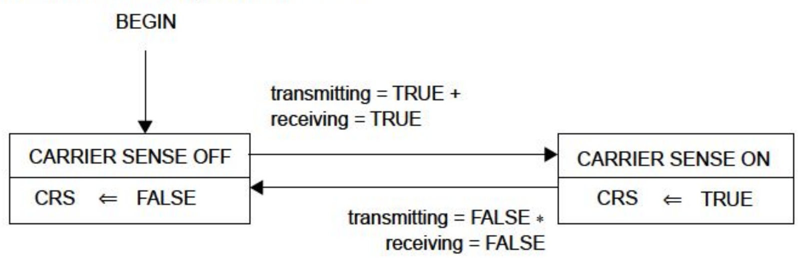


Figure 24-12—Carrier Sense state diagram

24.3 Physical Medium Attachment (PMA) sublayer

24.3.1 Service interface

The following specifies the service interface provided by the PMA to the PCS or another client, such as a repeater. These services are described in an abstract manner and do not imply any particular implementation.

The PMA Service Interface supports the exchange of code-bits between the PCS and/or Repeater entities. The PMA converts code-bits into NRZI format and passes these to the PMD, and vice versa. It also generates additional status indications for use by its client.

The following primitives are defined:

PMA_TYPE.indicate

PMA_UNITDATA.request

PMA_UNITDATA.indicate

PMA_CARRIER.indicate

PMA_LINK.indicate

PMA_LINK request

PMA_RXERROR.indicate

24.3.1.1 PMA_TYPE.indicate

This primitive is generated by the PMA to indicate the nature of the PMA instantiation. The purpose of this primitive is to allow clients to support connections to the various types of 100BASE-T PMA entities in a generalized manner.

24.3.1.1.1 Semantics of the service primitive

PMA_TYPE.indicate (pma_type)

The pma_type parameter for use with a 100BASE-X PMA is “X”.

24.3.1.1.2 When generated

The PMA continuously generates this primitive to indicate the value of pma_type.

24.3.1.1.3 Effect of receipt

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

24.3.1.2 PMA_UNITDATA.request

This primitive defines the transfer of data (in the form of code-bits) from the PMA's client to the PMA.

24.3.1.2.1 Semantics of the service primitive

PMA_UNITDATA.request (tx_code-bit)

This primitive defines the transfer of data (in the form of code-bits) from the PCS or other client to the PMA. The tx_code-bit parameter can take one of two values: ONE or ZERO.

24.3.1.2.2 When generated

The PCS or other client continuously sends, at a nominal 125 Mb/s rate, the appropriate code-bit for transmission on the medium.

24.3.1.2.3 Effect of receipt

Upon receipt of this primitive, the PMA generates a PMD_UNITDATA request primitive, requesting transmission of the indicated code-bit, in NRZI format (tx_nrzi-bit), on the MDI.

24.3.1.3 PMA_UNITDATA.indicate

This primitive defines the transfer of data (in the form of code-bits) from the PMA to the PCS or other client.

24.3.1.3.1 Semantics of the service primitive

PMA_UNITDATA.indicate (rx_code-bit)

The data conveyed by PMA_UNITDATA.indicate is a continuous code-bit sequence at a nominal 125 Mb/s rate. The rx_code-bit parameter can take one of two values: ONE or ZERO.

24.3.1.3.2 When generated

The PMA continuously sends code-bits to the PCS or other client corresponding to the PMD_UNITDATA.indicate primitives received from the PMD.

24.3.1.3.3 Effect of receipt

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

24.3.1.4 PMA_CARRIER.indicate

This primitive is generated by the PMA to indicate that a non-squelched, non-IDLE code-bit sequence is being received from the PMD. The purpose of this primitive is to give clients the earliest reliable indication of activity on the underlying continuous-signaling channel.

24.3.1.4.1 Semantics of the service primitive

PMA_CARRIER.indicate (carrier_status)

The carrier_status parameter can take on one of two values, ON or OFF, indicating whether a non-squelched, non-IDLE code-bit sequence (that is, carrier) is being received (ON) or not (OFF).

24.3.1.4.2 When generated

The PMA generates this primitive to indicate a change in the value of carrier_status.

24.3.1.4.3 Effect of receipt

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

24.3.1.5 PMA_LINK.indicate

This primitive is generated by the PMA to indicate the status of the underlying PMD receive link.

24.3.1.5.1 Semantics of the service primitive

PMA_LINK.indicate (link_status)

The link_status parameter can take on one of three values: READY, OK, or FAIL, indicating whether the underlying receive channel is intact and ready to be enabled by Auto-Negotiation (READY), intact and enabled (OK), or not intact (FAIL). Link_status is set to FAIL when the PMD sets signal_status to OFF; when Auto-Negotiation (optional) sets link_control to DISABLE; or when Far-End Fault Detect (optional) sets faulting to TRUE. When link_status ≠ OK, then rx_code-bit and carrier_status are undefined.

24.3.1.5.2 When generated

The PMA generates this primitive to indicate a change in the value of link_status.

24.3.1.5.3 Effect of receipt

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

24.3.1.6 PMA_LINK.request

This primitive is generated by the Auto-Negotiation algorithm, when implemented, to allow it to enable and disable operation of the PMA. See clause 28. When Auto-Negotiation is not implemented, the primitive is never invoked and the PMA behaves as if link_control = ENABLE.

24.3.1.6.1 Semantics of the service primitive

PMA_LINK request (link_control)

The link_control parameter takes on one of three values: SCAN_FOR_CARRIER, DISABLE, or ENABLE. Auto-Negotiation sets link_control to SCAN_FOR_CARRIER prior to receiving any fast link pulses, permitting the PMA to sense a 100BASE-X signal. Auto-Negotiation sets link_control to DISABLE when it senses an Auto-Negotiation partner (fast link pulses) and must temporarily disable the 100BASE-X PHY while negotiation ensues. Auto-Negotiation sets link_control to ENABLE when full control is passed to the 100BASE-X PHY.

24.3.1.6.2 When generated

Auto-Negotiation generates this primitive to indicate a change in link_control as described in clause 28.

24.3.1.6.3 Effect of receipt

This primitive affects operation of the PMA Link Monitor function as described in 24.3.4.4.

24.3.1.7 PMA_RXERROR.indicate

This primitive is generated by the PMA to indicate that an error has been detected during a carrier event.

24.3.1.7.1 Semantics of the service primitive

PMA_RXERROR.indicate (rxerror_status)

The rxerror_status parameter can take on one of two values: ERROR or NO_ERROR, indicating whether the received carrier event contains a detectable error (ERROR) or not (NO_ERROR). A carrier event is considered to be in error when it is not started by a Start-of-Stream Delimiter.

24.3.1.7.2 When generated

The PMA generates this primitive whenever a new, non-squelched carrier event is not started by a Start-of-Stream Delimiter.

24.3.1.7.3 Effect of receipt

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

24.3.2 Functional requirements

The 100BASE-X PMA comprises the following functions:

a) Mapping of transmit and receive code-bits between the PMA Service Interface and the PMD Service Interface;
b) Link Monitor, which maps the PMD_SIGNAL.indicate primitive to the PMA_LINK.indicate primitive, indicating the availability of the underlying PMD;
c) Carrier Detection, which generates the PMA_CARRIER.indicate and PMA_RXERROR.indicate primitives from inspection of received PMD signals; and
d) Far-End Fault (optional), comprised of the Far-End Fault Generate and Far-End Fault Detect processes, which sense receive channel failures and send the Far-End Fault Indication, and sense the Far-End Fault Indication.

Figure 24-4 includes a functional block diagram of the PMA.

24.3.2.1 Far-End fault

Auto-Negotiation provides a Remote Fault capability useful for detection of asymmetric link failures; i.e., channel error conditions detected by the far-end station but not the near-end station. Since Auto-Negotiation is specified only for media supporting eight-pin modular connectors, such as used by 100BASE-TX over unshielded twisted pair, Auto-Negotiation's Remote Fault capability is unavailable to other media for which it may be functionally beneficial, such as 100BASE-TX over shielded twisted pair or 100BASE-FX. A remote fault capability for 100BASE-FX is particularly useful due to this medium's applicability over longer distances (making end-station checking inconvenient) and for backbones (in which detection of link failures can trigger redundant systems).

For these reasons, 100BASE-X provides an optional Far-End Fault facility when Auto-Negotiation cannot be used. Far-End Fault shall not be implemented for media capable of supporting Auto-Negotiation.

When no signal is being received, as indicated by the PMD's signal detect function, the Far-End Fault feature permits the station to transmit a special Far-End Fault Indication to its far-end peer. The Far-End Fault Indication is sent only when a physical error condition is sensed on the receive channel. In all other situations, including reception of the Far-End Fault Indication itself, the PMA passes through tx_code-bit. (Note that the Far-End Fault architecture is such that IDLEs are automatically transmitted when the Far-End Fault Indication is detected. This is necessary to re-establish communication when the link is repaired.)

The Far-End Fault Indication is comprised of three or more repeating cycles, each of 84 ONEs followed by a single ZERO. This signal is sent in-band and is readily detectable but is constructed so as to not satisfy the 100BASE-X carrier sense criterion. It is therefore transparent to the PMA's client and to stations not implementing Far-End Fault.

As shown in figure 24-4, Far-End Fault is implemented through the Far-End Fault Generate, Far-End Fault Detect and the Link Monitor processes.

The Far-End Fault Generate process, which is interposed between the incoming tx_code-bit stream and the TX process, is responsible for sensing a receive channel failure (signal_status=OFF) and transmitting the Far-End Fault Indication in response. The transmission of the Far-End Fault Indication may start or stop at any time depending only on signal_status.

The Far-End Fault Detect process continuously monitors rx_code-bits from the RX process for the Far-End Fault Indication. Detection of the Far-End Fault Indication disables the station by causing the Link Monitor process to deassert link_status, which in turn causes the station to source IDLEs. Far-End Fault detection can also be used by management functions not specified in this clause.

24.3.2.2 Comparison to previous 802.3 PMAs

Previous 802.3 PMA's perform the additional functions of SQE Test and Jabber. Neither of these functions is implemented in the 100BASE-X PMA.

SQE Test is provided in other Physical Layers to check the integrity of the Collision Detection mechanism independently of the Transmit and Receive capabilities of the Physical Layer. Since 100BASE-X effects collision detection by sensing receptions that occur during transmissions, collision detection is dependent on the health of the receive channel. By checking the ability to properly receive signals from the PMD, the Link Monitor function therefore functionally subsumes the functions previously implemented by SQE Test.

The Jabber function prevents a DTE from causing total network failure under certain classes of faults. When using mixing media (e.g., coaxial cables or passive optical star couplers), this function must naturally be implemented in the DTE. 100BASE-X requires the use of an active repeater, with one DTE or repeater attached to each port. As an implementation optimization, the Jabber function has therefore been moved to the repeater in 100BASE-X.

24.3.3 State variables

24.3.3.1 Constants

FEF_CYCLES

The number of consecutive cycles (of FEF_ONES ONEs and a single ZERO) necessary to indicate the Far-End Fault Indication. This value is 3.

FEF_ONES

The number of consecutive ONEs to be transmitted for each cycle of the Far-End Fault Indication. This value is 84.

24.3.3.2 Variables

carrier_status

The carrier_status parameter to be communicated by the Carrier Detect process through the PMA_CARRIER.indicate primitive. Carrier is defined as receipt of 2 noncontiguous ZEROes in 10 code-bits.

Values: ON; carrier is being received
OFF; carrier is not being received

faulting

The faulting variable set by the Far-End Fault Detect process, when implemented, indicating whether or not a Far-End Fault Indication is being sensed. This variable is used by the Link Monitor process to force link_status to FAIL.When Far-End Fault is not implemented, this variable is always FALSE.

Values: TRUE; Far-End Fault Indication is being sensed
FALSE; Far-End Fault Indication is not being sensed

link_control

The link_control parameter as communicated by the PMA_LINK.request primitive. When Auto-Negotiation is not implemented, the value of link_control is always ENABLE. See clause 28 for a complete definition.

link_status

The link_status parameter as communicated by the Link Monitor process through the PMA_LINK.indicate primitive.

Values: FAIL; the receive channel is not intact
READY; the receive channel is intact and ready to be enabled by Auto-Negotiation
OK; the receive channel is intact and enabled for reception

r_bits [9:0]

In Carrier Detect, a vector of the 10 most recently received code-bits from the PMD RX process. r_bits [0] is the most recently received (newest) code-bit; r_bits [9] is the least recently received code-bit (oldest). r_bits is an internal variable used exclusively by the Carrier Detect process.

rx_code-bit

The rx_code-bit parameter as delivered by the RX process, which operates in synchronism with the PMD_UNITDATA.indicate primitive. rx_code-bit is the most recently received code-bit from the PMD after conversion from NRZI.

rxerror_status

The rxerror_status parameter to be communicated by the Carrier Detect process through the PMA_RXERROR.indicate primitive.

Values: NO_ERROR; no error detected in the carrier event being received
ERROR; the carrier event being received is in error

signal_status

The signal_status parameter as communicated by the PMD_SIGNAL.indicate primitive.

Values: ON; the quality and level of the received signal is satisfactory
OFF; the quality and level of the received signal is not satisfactory

tx_code-bit_in

In Link Fault Generate, the tx_code-bit parameter as conveyed to the PMA from the PMA client by the PMA_UNITDATA request.

tx_code-bit_out

In Link Fault Generate, the tx_code-bit parameter to be passed to the TX process. Note that this is called tx_code-bit by the TX process.

24.3.3.3 Functions

SHIFTLEFT (rx_bits)

In Carrier Detect, this function shifts rx_bits left one bit placing rx_bits [8] in rx_bits [9], rx_bits [7] in rx_bits [8] and so on until rx_bits [1] gets rx_bits [0].

24.3.3.4 Timers

stabilize_timer

An implementation-dependent delay timer between 330 µs and 1000 µs, inclusive, to ensure that the link is stable.

24.3.3.5 Counters

num_cycles

In Link Fault Detect, a counter containing the number of consecutive Far-End Fault cycles currently sensed. This counter gets reset on intialization or when the bit stream fails to qualify as a potential Far-End Fault Indication. It never exceeds FEF_CYCLES.

num_ones

This represents two separate and independent counters: In Link Fault Generate, a counter containing the number of consecutive ONEs already sent during this cycle of the Far-End Fault Indication. In Link Fault Detect, a counter containing the number of consecutive ONEs currently sensed; it gets reset whenever a ZERO is detected or when the bit stream fails to qualify as a potential Far-End Fault Indication. These counters never exceed FEF_ONES.

24.3.3.6 Messages

PMD_UNITDATA.indicate (rx_nrzi-bit)

A signal sent by the PMD signifying that the next nrzi-bit is available from the medium. nrzi-bit is converted (instantaneously) to code-bit by the RX process and used by the Carrier Detect process.

5xPMD_UNITDATA.indicates

In Carrier Detect, this shorthand notation represents repetition of the preceding state five times synchronized with five successive PMD_UNITDATA.indicates.

PMA_UNITDATA request (tx_code-bit)

A signal sent by the PMA's client signifying that the next nrzi-bit is available for transmission. For this process, the tx_code-bit parameter is interpreted as tx_code-bit_in.

24.3.4 Process specifications and state diagrams

24.3.4.1 TX

The TX process passes data from the PMA's client directly to the PMD. The PMA shall implement the TX process as follows: Upon receipt of a PMA_UNITDATA request (tx_code-bit), the PMA performs a conversion to NRZI format and generates a PMD_UNITDATA request (tx_nrzi-bit) primitive with the same logical value for the tx_nrzi-bit parameter. Note that tx_code-bit is equivalent to tx_code-bit_out of the Link Fault Generate process when implemented.

24.3.4.2 RX

The RX process passes data from the PMD directly to the PMA's client and to the Carrier Detect process. The PMA shall implement the RX process as follows: Upon receipt of a PMD_UNITDATA.indicate (rx_nrzi-bit), 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.

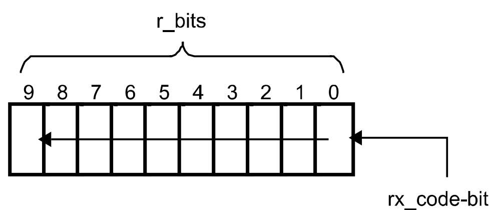


Figure 24-13—Carrier Detect reference diagram

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.

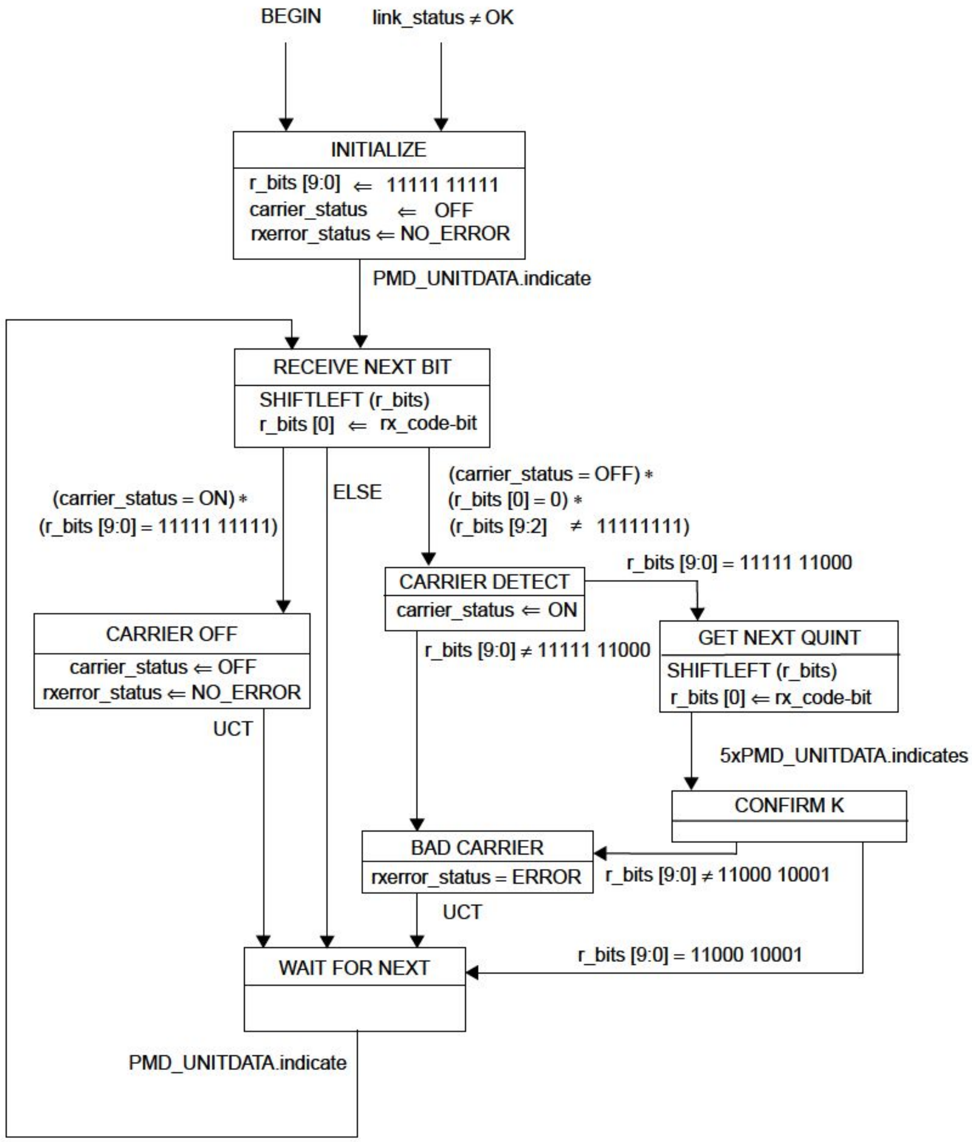
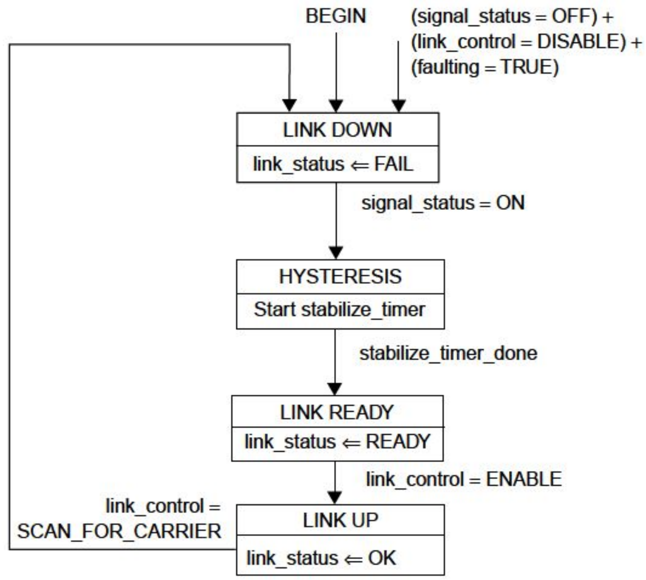


Figure 24-14—Carrier Detect state diagram

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:

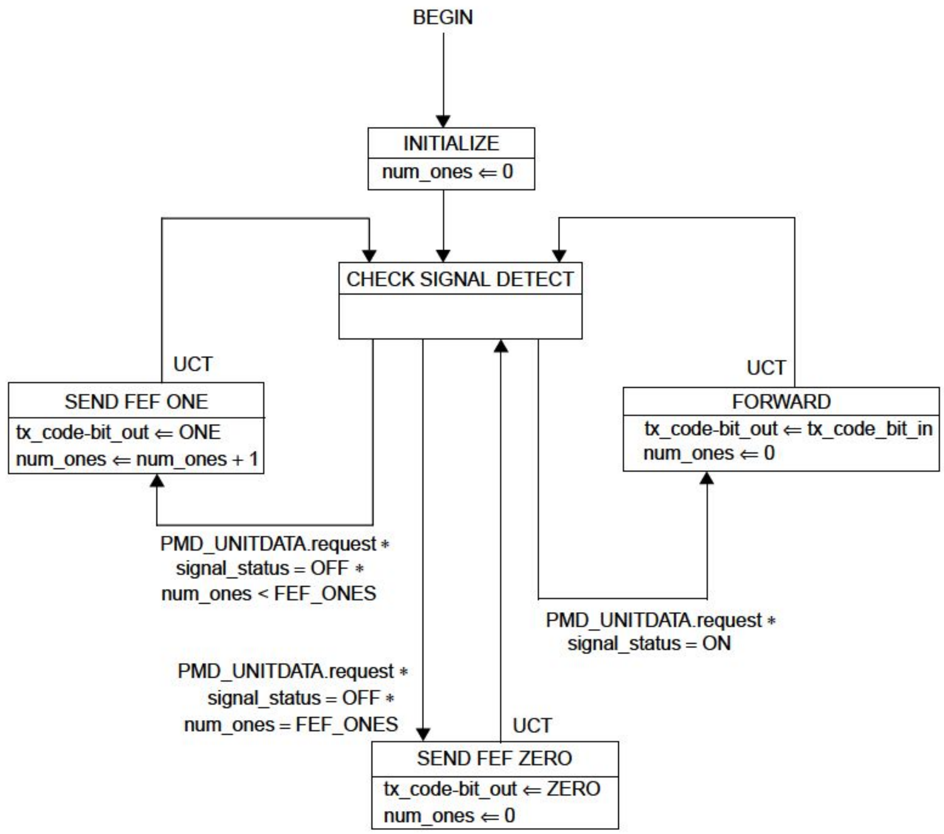


Figure 24-16—Far-End Fault Generate state diagram

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.

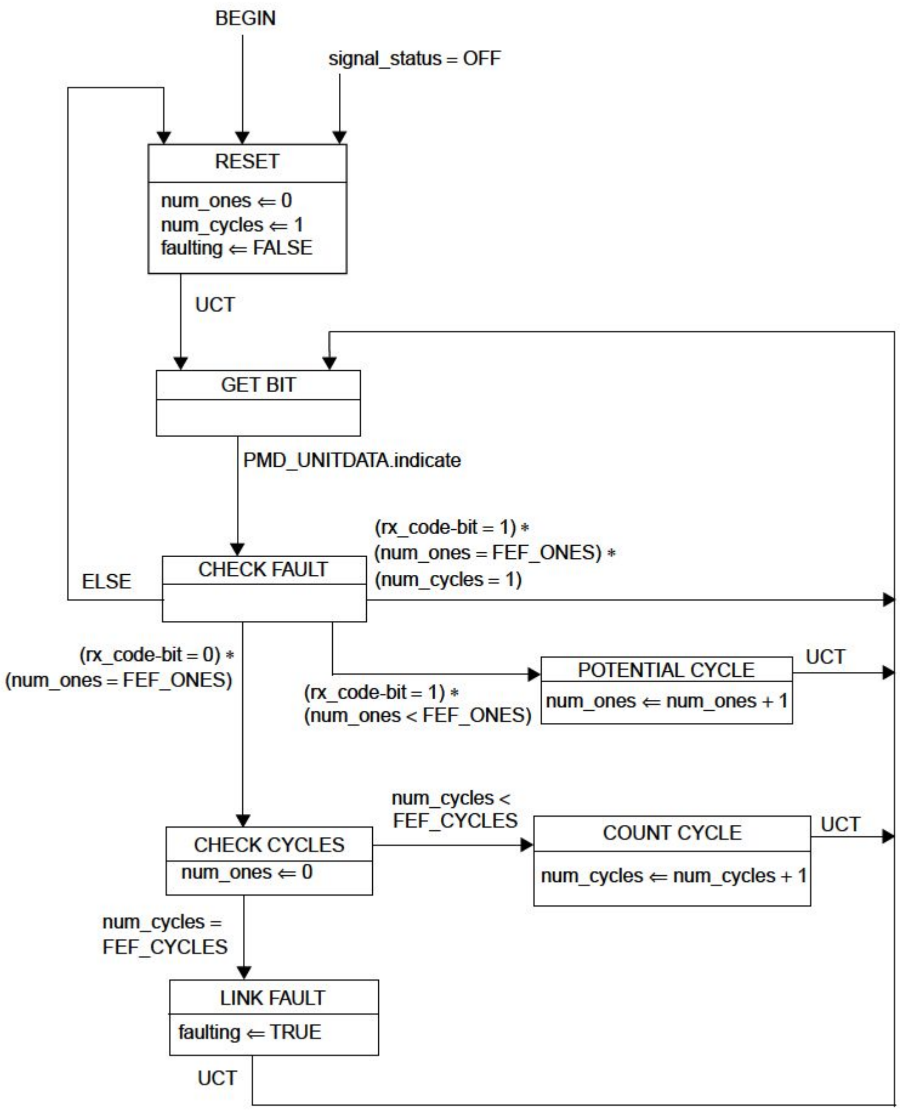


Figure 24-17—Far-End Fault Detect state diagram

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 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. 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 [ ] (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 DTE: M 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 REP: O		
TM4	Compliance with DTE bit delay constraints	24.6.2	DTE:M		
TM5	Compliance with Carrier De-assert/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: SM_PM_BYPASS request(Control_Action = Insert)
SM_PM_CONTROL request (Control_Action)	Assume: 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.