(5) Number of octetsTransmittedOK: Recommended, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).

(6) Number of Frames with deferredTransmissions: Recommended, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).

(7) Number of multicastFramesTransmittedOK: Optional, Read only, 32 bit counter.
This contains a count of frames that are successfully transmitted, as indicated by the status value transmitOK, to a group destination address other than broadcast. The update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).

(8) Number of broadcastFramesTransmittedOK: Optional, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).

5.2.2.1.2 MAC Transmit Error Statistics Descriptions. This section defines the MAC sublayer transmission related error statistics.

(1) Number of lateCollision: Recommended Read only, 32 bit counter.
This contains a count of the times that a collision has been detected later than 512 bit times 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 update is incremented in the LayerMgmtTransmitCounters procedure (5.2.4.2).

(2) Number of frames aborted due to excessiveCollision: Recommended, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).

(3) Number of frames lost due to internalMACTransmitError: Recommended, Read only, 32 bit counter.
This contains 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 section are incremented. The exact meaning and mechanism for incrementing this counter is implementation-dependent.

(4) Number of carrierSenseErrors: Recommended, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).

(5) Number of frames with excessiveDeferral: Optional, Read only, 32 bit counter.
This contains a count of frames that were 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 update occurs in the LayerMgmtTransmitCounters procedure (5.2.4.2).

5.2.2.1.3 MAC Receive Statistics Descriptions

(1) Number of framesReceivedOK: Mandatory, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

(2) Number of octetsReceivedOK: Recommended, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

(3) Number of multicastFramesReceivedOK: Optional, Read only, 32 bit counter.
This contains a count of frames that are successfully received and are directed to an active non-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 update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

(4) Number of broadcastFramesReceivedOK: Optional, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

5.2.2.1.4 MAC Receive Error Statistics Descriptions. This section defines the MAC sublayer reception related error statistics. Note that 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 following counters are primarily incremented based on the status returned to the LLC, and therefore the hierarchical order of the counters is determined by the order of the status.

(1) Number of frames received with frameCheckSequenceErrors: Mandatory, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

(2) Number of frames received with alignmentErrors: Mandatory, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

(3) Number of frames lost due to internalMACReceiveError: Recommended, Read only, 32 bit counter.
This contains 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 section are incremented. The exact meaning and mechanism for incrementing this counter is implementation-dependent.

(4) Number of frames received with inRangeLengthErrors: Optional, Read only, 32 bit counter.
This contains 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 update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

(5) Number of frames received with outOfRangeLengthField: Optional, Read only, 32 bit counter.
This contains a count of frames with a length field value greater than the maximum allowed LLC data size. The update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

(6) Number of frames received with frameTooLongErrors: Optional, Read only, 32 bit counter.
This contains a count of frames that are received and exceed the maximum permitted frame size. This counter is incremented when the status of a frame reception is frameTooLong. The update occurs in the LayerMgmtReceiveCounters procedure (5.2.4.3).

5.2.2.2 MAC Actions. This subsection defines the actions offered by the MAC sublayer to the LME client.

These actions enable the LME client to influence the behavior of the MAC sublayer, i.e., to execute "actions" on the MAC sublayer for management purposes. Many of the following actions enable or disable some function; if either the enable or disable action is implemented, the corresponding disable or enable action must also be implemented. If the enable/disable action is supported, then its corresponding read action must also be supported.

In implementing any of the following actions, receptions and transmissions that are in progress are completed before the action takes effect.

The security considerations related to the following actions should be properly addressed by the SMAE.

The items in parenthesis in the descriptions are the procedures that are affected by these actions.

5.2.2.2.1 MAC Action Definitions

(1) initializeMAC: Mandatory
Call the Initialize procedure (4.2.7.5). This action also results in the initialization of the PLS.

(2) enablePromiscuousReceive: Recommended
Cause the LayerMgmtRecognizeAddress function to accept frames regardless of their destination address (LayerMgmtRecognizeAddress function).
Frames without errors received solely because this action is set are counted as frames received correctly; frames received in this mode that do contain errors update the appropriate error counters.

(3) disablePromiscuousReceive: Recommended
Cause the MAC sublayer to return to the normal operation of carrying out address recognition procedures for station, broadcast, and multicast group addresses (LayerMgmtRecognizeAddress function).

(4) readPromiscuousStatus: Recommended
Return true if promiscuous mode enabled, and false otherwise (LayerMgmtRecognizeAddress function).

(5) addGroupAddress: Recommended
Add the supplied multicast group address to the address recognition filter (RecognizedAddress function).

(6) deleteGroupAddress: Recommended
Delete the supplied multicast group address from the address recognition filter (RecognizeAddress function).

(7) readMulticastAddressList: Recommended
Return the current multicast address list.

(8) enableMacSublayer: Optional
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.

(9) disableMacSublayer: Optional
Cause 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.

(10) readMACEnableStatus: Optional
Return true if MAC sublayer is enabled, and false if disabled. This is accomplished by checking the values of the receiveEnabled and transmitEnabled variables.

(11) enableTransmit: Optional
Enable MAC sublayer frame transmission (TransmitFrame function). This is accomplished by setting transmitEnabled to true.

(12) disableTransmit: Optional
Inhibit the transmission of further frames by the MAC sublayer (TransmitFrame function). This is accomplished by setting transmitEnabled to false.

(13) readTransmitEnableStatus: Optional
Return true if transmission is enabled and false otherwise. This is accomplished by checking the value of the transmitEnabled variable.

(14) enableMulticastReceive: Optional
Cause the MAC sublayer to return to the normal operation of multicast frame reception.

(15) disableMulticastReceive: Optional
Inhibit the reception of further multicast frames by the MAC sublayer.

(16) readMulticastReceiveStatus: Optional
Return true if multicast receive is enabled, and false otherwise.
(17) modifyMACAddress: Optional
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.
(18) readMACAddress: Optional
Read the current MAC station address.
(19) executeSelftest: Optional
Execute a self test and report the results (success or failure). The mechanism employed to carry out the self test is not defined in this standard.

5.2.3 Physical Layer Management Facilities. This section of the standard defines the Layer Management facilities for the Physical Layer.

5.2.3.1 Physical Statistics. The statistics defined in this section are implemented by means of counters.

In the following definition, the term "Read only" specifies that the object cannot be written by the client of the LME.

Note that the carrierSenseFailed statistic is a statistic relating to the physical layer, but is listed and maintained in the MAC sublayer for ease of implementation.

5.2.3.1.1 Physical Statistics Descriptions

(1) Number of SQETestErrors: Recommended, Read only, 32^8 bit counter.
This contains 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).

5.2.4 Layer Management Model. The following model provides the descriptions for Layer Management facilities.

5.2.4.1 Common Constants and Types. The following are the common constants and types required for the Layer Management procedures:

```
const
   maxFrameSize = ...; {in octets, implementation-dependent, see 4.4}
   maxDeferTime = ...; {2 × (maxFrameSize × 8), in bits, error timer limit for maxDeferTime}
   maxLarge = 4294967295; {maximum value (2^32 – 1) of wraparound 32 bit counter}
   max64 = xxxxxxxxx; {maximum value (2^64 – 1) of wraparound 64 bit counter}
   oneBitTime = 1; {the period it takes to transmit one bit}
type
   CounterLarge = 0..maxLarge--See footnote.;
```

5.2.4.2 Transmit Variables and Procedures. The following items are specific to frame transmission:

```
var
   excessDefer: Boolean; {set in process DeferTest}
   carrierSenseFailure: Boolean; {set in process CarrierSenseTest}
   transmitEnabled: Boolean; {set by MAC action}
   lateCollisionError: Boolean; {set in Section 4 procedure WatchForCollision}
   deferred: Boolean; {set in Section 4 function TransmitLinkMgmt}
   carrierSenseTestDone: Boolean; {set in process CarrierSenseTest}
```

[8]32 bit counter size specification is not a part of this ISO/IEC standard. Resolution of 32 vs. 64 bit counter size will be addressed during the further work required to develop this section into a specification sufficient for ISO/IEC interoperability requirements.

```
lateCollisionCount: 0..attemptLimit – 1; {count of late collision that is used in Section 4
                                          TransmitLinkMgmt}

{MAC transmit counters}
framesTransmittedOK: CounterLarge; {mandatory}
singleCollisionFrames: CounterLarge; {mandatory}
multipleCollisionFrames: CounterLarge; {mandatory}
collisionFrames: array [1..attemptLimit – 1] of CounterLarge; {recommended}
octetsTransmittedOK: CounterLarge; {recommended}
deferredTransmissions: CounterLarge; {recommended}
multicastFramesTransmittedOK: CounterLarge; {optional}
broadcastFramesTransmittedOK: CounterLarge; {optional}
{MAC transmit error counters}
lateCollision: CounterLarge; {recommended}
excessiveCollision: CounterLarge; {recommended}
carrierSenseErrors: CounterLarge; {optional}
excessiveDeferral: CounterLarge; {optional}
```

Procedure LayerMgmtTransmitCounters is invoked from the TransmitLinkMgmt function in 4.2.8 to update the transmit and transmit error counters.

```
procedure LayerMgmtTransmitCounters;
begin
  while not carrierSenseTestDone do nothing;
  if transmitSucceeding then
  begin
    IncLargeCounter(framesTransmittedOK);
    SumLarge(octetsTransmittedOK, dataSize/8); {dataSize (in bits) is defined in 4.2.7.1}
    if destinationField = ... {check to see if to a multicast destination}
      then IncLargeCounter(multicastFramesTransmittedOK);
    if destinationField = ... {check to see if to a broadcast destination}
      then IncLargeCounter(broadcastFramesTransmittedOK);

    if attempts > 1 then
    begin {transmission delayed by collision}
      if attempts = 2 then
        IncLargeCounter(singleCollisionFrames) {delay by 1 collision}
      else {attempts > 2, delayed by multiple collisions}
        IncLargeCounter(multipleCollisionFrames)
      IncLargeCounter(collisionFrames[attempts – 1]);
    end; {delay by collision}
  end; {transmitSucceeding}

  if deferred and (attempts = 1) then
    IncLargeCounter(deferredTransmissions);
  if lateCollisionCount > 0 then {test if late collision detected}
    SumLarge(lateCollision, lateCollisionCount);
  if attempts = attemptLimit and not transmitSucceeding then
    IncLargeCounter(excessiveCollision);
  if carrierSenseFailure then
    IncLargeCounter(carrierSenseErrors);
  if excessDefer then
    IncrementLargeCounter(excessiveDeferral);
end; {LayerMgmtTransmitCounters}
```

The DeferTest process sets the excessDefer flag if a transmission attempt has been deferred for a period of time longer than maxDeferTime.

```
process DeferTest;
   var deferBitTimer: 0..maxDeferTime;
begin
   cycle
   begin
      deferCount := 0;
      while frameWaiting and not excessDefer do
      begin
         Wait(oneBitTime); {see 4.3.3}
         if deferBitTimer = maxDeferTime then
            excessDefer := true
         else
            deferBitTimer := deferBitTimer + 1;
      end; {while}
      while transmitting do nothing;
   end; {cycle}
end; {DeferTest}
```

The CarrierSenseTest process sets the carrierSpenseFailure flag if carrier sense disappears while transmitting or if it never appears during an entire transmission.

```
process CarrierSenseTest;
   var
      carrierSeen: Boolean; {Running indicator of whether or not carrierSense has been true at any
                             time during the current transmission}
      collisionSeen: Boolean; {Running indicator of whether or not the collisionDetect asserted any
                               time during the entire transmission}
begin
   cycle {main loop}
      while not transmitting do nothing; {wait for start of transmission}
      carrierSenseFailure := false;
      carrierSeen := false;
      collisionSeen := false;
      carrierSenseTestDone := false;
      while transmitting do
      begin {inner loop}
         if carrierSense then
            carrierSeen := true;
         else
            if carrierSeen then {carrierSense disappeared before end of transmission}
               carrierSenseFailure := true;
         if collisionDetect then
            collisionSeen := true;
      end; {inner loop}
      if not carrierSeen then
         carrierSenseFailure := true {carrier sense never appeared}
      else
         if collisionSeen then
            carrierSenseFailure := false;
      carrierSenseTestDone := true;
   end; {main loop}
end; {CarrierSenseTest}
```

5.2.4.3 Receive Variables and Procedures. The following items are specific to frame reception:

```
var
   receiveEnabled: Boolean; {set by MAC action}
```

```
{MAC receive counters}
framesReceivedOK: CounterLarge; {mandatory}
octetsReceivedOK: CounterLarge; {recommended}

{MAC receive error counters}
frameCheckSequenceErrors: CounterLarge; {mandatory}
alignmentErrors: CounterLarge; {mandatory}
inRangeLengthErrors: CounterLarge; {optional}
outOfRangeLengthField: CounterLarge; {optional}
frameTooLongErrors: CounterLarge; {optional}

{MAC receive address counters}
multicastFramesReceivedOK: CounterLarge; {optional}
broadcastFramesReceivedOK: CounterLarge; {optional}
```

Procedure LayerMgmtReceiveCounters is called by ReceiveLinkMgmt in 4.2.9 and increments the appropriate receive counters.

```
procedure LayerMgmtReceiveCounters (status: ReceiveStatus);
begin
  case status of
    receiveDisabled:
    begin
      nothing;
    end {receiveDisabled}
    receiveOK:
    begin
      IncLargeCounter(framesReceivedOK);
      SumLarge(octetsReceivedOK, dataSize/8); {dataSize (in bits) is defined in 4.2.7.1}
      if destinationField = ... {check to see if to a multicast destination}
        then IncLargeCounter(multicastFramesReceivedOK);
      if destinationField = ... {check to see if to a broadcast destination}
        then IncLargeCounter(broadcastFramesReceivedOK);
    end; {receiveOK}
    frameTooLong:
    begin
      IncLargeCounter(frameTooLongErrors);
    end; {frameTooLong}
    frameCheckError:
    begin
      IncLargeCounter(frameCheckSequenceErrors);
    end; {frameCheckError}
    alignmentError:
    begin
      IncLargeCounter(alignmentErrors);
    end; {alignmentError}
    lengthError:
    begin
      if {length field value is between the minimum unpadded LLCDataSize and maximum allowed
      LLCDataSize inclusive, and does not match the number of LLC data octets received} or
      {length field value is less than the minimum allowed unpadded LLC data size and the number
      of LLC data octets received is greater than the minimum unpadded LLCDataSize} then
      IncLargeCounter(inRangeLengthError);
    end; {lengthError}
  end; {case status}
  if {length field value is greater than the maximum allowed LLCDataSize} then
  IncLargeCounter(outOfRangeLengthField);
```

```
end; {LayerMgmtReceiveCounters}
```

Function LayerMgmtRecognizeAddress checks if reception of certain addressing types has been enabled. Note that in Pascal, assignment to a function causes the function to return immediately.

```
function LayerMgmtRecognizeAddress(address: AddressValue): Boolean;
begin
   if {promiscuous receive enabled} then
      LayerMgmtRecognizeAddress := true;
   if address = ... {MAC station address} then
      LayerMgmtRecognizeAddress := true;
   if address = ... {broadcast address} then
      LayerMgmtRecognizeAddress := true;
   if address = ... {one of the addresses on the multicast list and multicast reception is enabled} then
      LayerMgmtRecognizeAddress := true;
   LayerMgmtRecognizeAddress := false;
end; {LayerMgmtRecognizeAddress}
```

5.2.4.4 Common Procedures. Procedure LayerMgmtInitialize initializes all the variables and constants required to implement Layer Management.

```
procedure LayerMgmtInitialize;
begin
   {initialize flags for enabling/disabling transmission and reception}
   receiveEnabled := true;
   transmitEnabled := true;

   {initialize transmit flags for DeferTest and CarrierSenseTest}
   deferred := false;
   lateCollisionError := false;
   excessDefer := false;
   carrierSenseFailure := false;
   carrierSenseTestDone := false;

   {Initialize all MAC sublayer management counters to zero}

end; {LayerMgmtInitialize}
```

Procedure IncLargeCounter increments a 32 bit wraparound counter.

```
procedure IncLargeCounter (var counter: CounterLarge);
begin
   {increment the 32 bit counter}
end; {IncLargeCounter}
```

Procedure SumLarge adds a value to a 32 bit wraparound counter.

```
procedure SumLarge (
   var counter: CounterLarge;
   var offset: Integer);
begin
   {add offset to the 32 bit counter}
end; {SumLarge}
```

6. PLS Service Specifications

6.1 Scope and Field of Application. This section specifies the services provided by the Physical Signaling (PLS) sublayer to the MAC sublayer for the CSMA/CD section of the Local Area Network Standard, Fig 6-1. The services are described in an abstract way and do not imply any particular implementation.

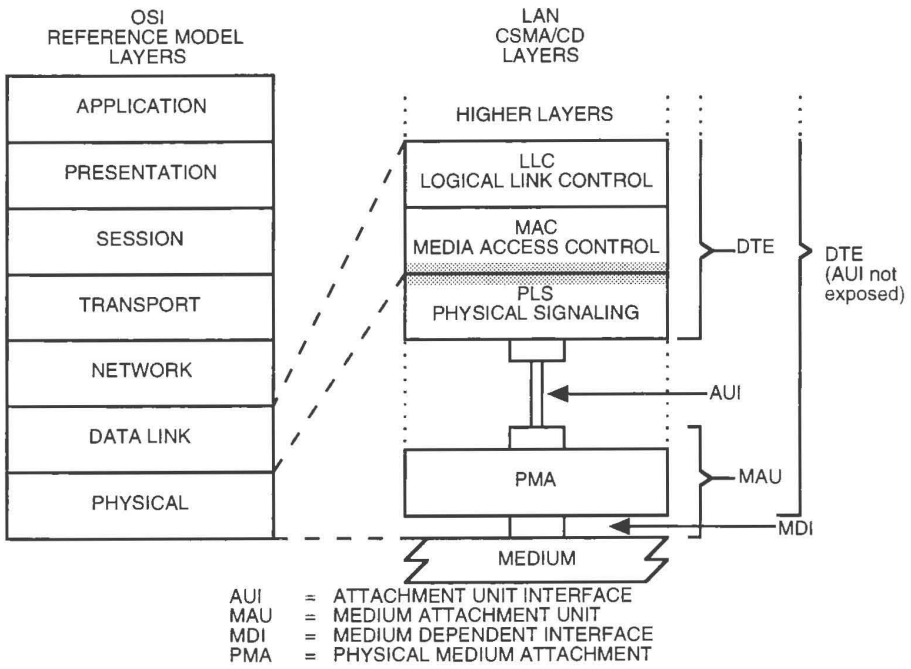


Fig 6-1
Service Specification Relationship to the IEEE 802.3 CSMA/CD LAN Model

6.2 Overview of the Service

6.2.1 General Description of Services Provided by the Layer. The services provided by the PLS sublayer allow the local MAC sublayer entity to exchange data bits (PLS data_units) with peer MAC sublayer entities.

6.2.2 Model Used for the Service Specification. The model used in this service specification is identical to that used in 1.2.2.1.

6.2.3 Overview of Interactions. The primitives associated with the MAC sublayer to PLS sublayer interface fall into two basic categories:

(1) Service primitives that support MAC peer-to-peer interactions
(2) Service primitives that have local significance and support sublayer-to-sublayer interactions

The following primitives are grouped into these two categories:

(1) Peer-to-Peer
PLS_DATA.request
PLS_DATA.indication

(2) Sublayer-to-Sublayer
PLS_CARRIER.indication
PLS_SIGNAL.indication

The PLS_DATA primitives support the transfer of data from a single MAC sublayer entity to all other peer MAC sublayer entities contained within the same local area network defined by the broadcast medium.

NOTE: This also means that all bits transferred from a given MAC sublayer entity will in turn be received by the entity itself.

The PLS_CARRIER and the PLS_SIGNAL primitives provide information needed by the local MAC sublayer entity to perform the media access functions.

6.2.4 Basic Services and Options. All of the service primitives described in this section are considered mandatory.

6.3 Detailed Service Specification

6.3.1 Peer-to-Peer Service Primitives

6.3.1.1 PLS_DATA.request

6.3.1.1.1 Function. This primitive defines the transfer of data from the MAC sublayer to the local PLS entity.

6.3.1.1.2 Semantics of the Service Primitive. The primitive shall provide the following parameters:

PLS_DATA.request (OUTPUT_UNIT)

The OUTPUT_UNIT parameter can take on one of three values: ONE, ZERO, or DATA_COMPLETE and represent a single data bit. The DATA_COMPLETE value signifies that the Media Access Control sublayer has no more data to output.

6.3.1.1.3 When Generated. This primitive is generated by the MAC sublayer to request the transmission of a single data bit on the physical medium or to stop transmission.

6.3.1.1.4 Effect of Receipt. The receipt of this primitive will cause the PLS entity to encode and transmit either a single data bit or to cease transmission.

6.3.1.2 PLS_DATA.indicate

6.3.1.2.1 Function. This primitive defines the transfer of data from the PLS sublayer to the MAC sublayer.

6.3.1.2.2 Semantics of the Service Primitive. The semantics of the primitive are as follows:

PLS_DATA.indicate (INPUT_UNIT)

The INPUT_UNIT parameter can take one of two values each representing a single bit: ONE or ZERO.

6.3.1.2.3 When Generated. The PLS_DATA.indicate is generated to all MAC sublayer entities in the network after a PLS_DATA.request is issued.

NOTE: An indicate is also presented to the MAC entity that issued the request.

6.3.1.2.4 Effect of Receipt. The effect of receipt of this primitive by the MAC sublayer is unspecified.

6.3.2 Sublayer-to-Sublayer Service Primitives

6.3.2.1 PLS_CARRIER.indicate

6.3.2.1.1 Function. This primitive transfers the status of the activity on the physical medium from the PLS sublayer to the MAC sublayer.

6.3.2.1.2 Semantics of the Service Primitive. The semantics of the primitive are as follows:

PLS_CARRIER.indicate (CARRIER_STATUS)

The CARRIER_STATUS parameter can take one of two values: CARRIER_ON or CARRIER_OFF. The CARRIER_ON value indicates that the DTE Physical Layer had received an *input* message or a *signal_quality_error* message from the MAU. The CARRIER_OFF value indicates that the DTE Physical Layer had received an *input_idle* message and is not receiving an SQE *signal_quality_error* message from the MAU.

6.3.2.1.3 When Generated. The PLS_CARRIER.indicate service primitive is generated whenever CARRIER_STATUS makes a transition from CARRIER_ON to CARRIER_OFF or vice versa.

6.3.2.1.4 Effect of Receipt. The effect of receipt of this primitive by the MAC sublayer is unspecified.

6.3.2.2 PLS_SIGNAL.indicate

6.3.2.2.1 Function. This primitive transfers the status of the Physical Layer signal quality from the PLS sublayer to the MAC sublayer.

6.3.2.2.2 Semantics of the Service Primitive. The semantics of the service primitive are as follows:

PLS_SIGNAL.indicate (SIGNAL_STATUS)

The SIGNAL_STATUS parameter can take one of two values: SIGNAL_ERROR or NO_SIGNAL_ERROR. The SIGNAL_ERROR value indicates to the MAC sublayer that the PLS has received a signal_quality_error message from the MAU. The NO_SIGNAL_ERROR value indicates that the PLS has ceased to receive signal_quality_error messages from the MAU.

6.3.2.2.3 When Generated. The PLS_SIGNAL.indicate service primitive is generated whenever SIGNAL_ STATUS makes a transition from SIGNAL_ERROR to NO_SIGNAL_ERROR or vice versa.

6.3.2.2.4 Effect of Receipt. The effect of receipt of this primitive by the MAC sublayer is unspecified.

7. Physical Signaling (PLS) and Attachment Unit Interface (AUI) Specifications

7.1 Scope. This section defines the logical, electrical, and mechanical characteristics for the PLS and AUI between Data Terminal Equipment and Medium Attachment Units used in CSMA/CD local area networks. The relationship of this specification to the entire ISO [IEEE] Local Area Network standards is shown in Fig 7-1. The purpose of this interface is to provide an interconnection that is simple and inexpensive and that permits the development of simple and inexpensive MAUs.

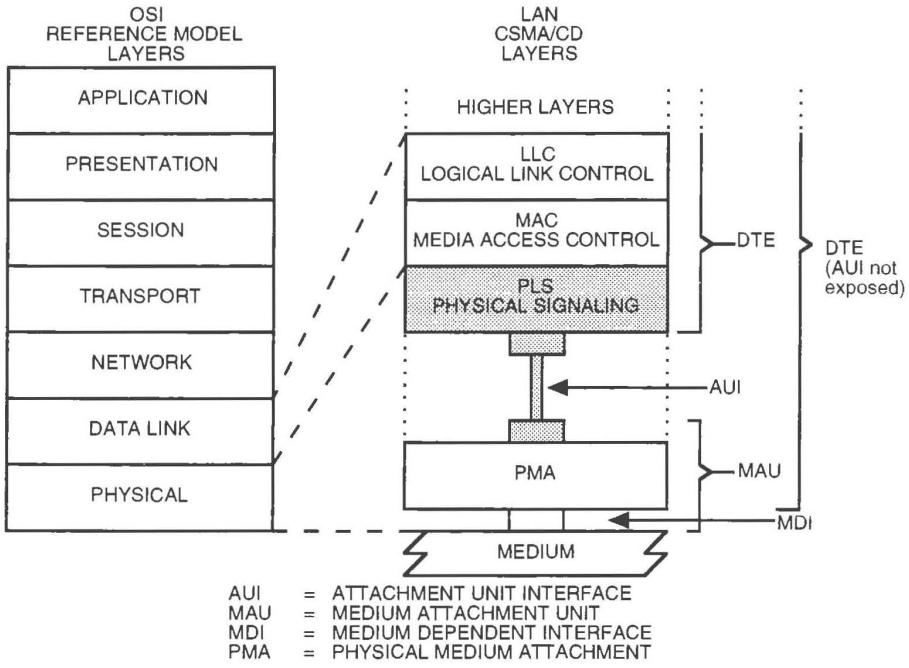


Fig 7-1
Physical Layer Partitioning, Relationship to the ISO Open Systems Interconnection (OSI) Reference Model

This interface has the following characteristics:

(1) Capable of supporting one or more of the specified data rates
(2) Capable of driving up to 50 m (164 ft) of cable
(3) Permits the DTE to test the AUI, AUI cable, MAU, and the medium itself
(4) Supports MAUs for baseband coax, broadband coax, and baseband fiber

7.1.1 Definitions

Attachment Unit Interface (AU Interface) (AUI). In a local area network, the interface between the medium attachment unit and the data terminal equipment within a data station.

NOTE: The AUI carries encoded control and data signals between the DTE's PLS sublayer and the MAU's PMA sublayer and provides for duplex data transmission.

BR. The rate of data throughput (bit rate) on the medium in bits per second.

bit time. The duration of one bit symbol (1/BR).

circuit. The physical medium on which signals are carried across the AUI. The data and control circuits consist of an A circuit and a B circuit forming a balanced transmission system so that the signal carried on the B circuit is the inverse of the signal carried on the A circuit.

Clocked Data One (CD1). A Manchester encoded data "1." A CD1 is encoded as a LO for the first half of the bit-cell and a HI for the second half of the bit-cell.

Clocked Data Zero (CD0). A Manchester encoded data "0." A CD0 is encoded as a HI for the first half of the bit-cell and a LO for the second half of the bit-cell.

Control Signal One (CS1). An encoded control signal used on the Control In and Control Out circuits. A CS1 is encoded as a signal at half the bit rate (BR/2).

Control Signal Zero (CS0). An encoded control signal used on the Control In and Control Out circuits. A CS0 is encoded as a signal at the bit rate (BR).

idle (IDL). A signal condition where no transition occurs on the transmission line is used to define the end of a frame and ceases to exist after the next LO to HI transition on the AUI circuits. An IDL always begins with a HI signal level. A driver is required to send the IDL signal for at least 2 bit times and a receiver is required to detect IDL within 1.6 bit times. See 7.3 for additional details.

7.1.2 Summary of Major Concepts

(1) Each direction of data transfer is serviced with two (making a total of four) balanced circuits: "Data" and "Control."
(2) The Data and Control circuits are independently self-clocked, thereby, eliminating the need for separate timing circuits. This is accomplished with encoding of all signals. The Control circuit signaling rate is nominally (but not of necessity exactly) equal to the Data circuit signaling rate.
(3) The Data circuits are used only for data transfer. No control signals associated with the interface are passed on these circuits. Likewise, the Control circuits are used only for control message transfer. No data signals associated with the interface are passed on these circuits.

7.1.3 Application. This standard applies to the interface used to interconnect Data Terminal Equipment (DTE) to a MAU that is not integrated as a physical part of the DTE. This interface is used to

(1) Provide the DTE with media independence for baseband coax, broadband coax, and baseband fiber media so that identical PLS, MAC and LLC may be used with any of these media.
(2) Provide for the separation by cable of up to 50 m (164 ft) the DTE and the MAU.

7.1.4 Modes of Operation. The AUI can operate in two different modes. All interfaces shall support the normal mode. The monitor mode is optional.

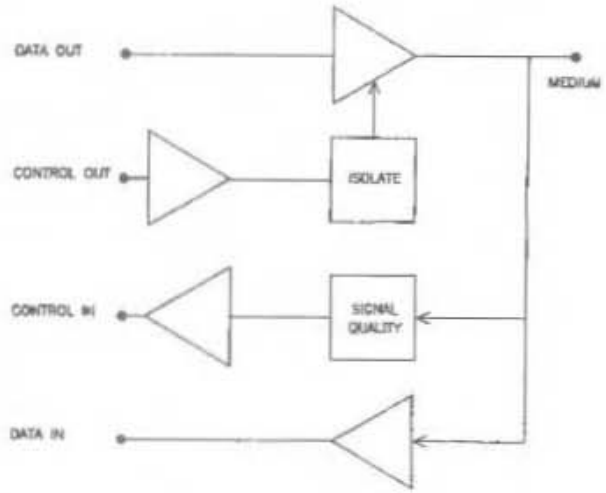
When the interface is being operated in the *normal* mode, the AUI is logically connected to the MDI. The DTE is required to follow the media access algorithms, which provide a single access procedure compatible with all local area network media, to send data over the AUI. The MAU always sends back to the DTE whatever data the MAU receives on the MDI.

When the interface is in the optional *monitor* mode, the MAUs transmitter is logically isolated from the medium. The MAU, in this mode, functions as an observer on the medium. Both the input function and the signal quality error function are operational (see the MAU state diagrams for specific details).

7.1.5 Allocation of Function. The allocation of functions in the AUI is such that the majority of the functionality required by the interface can be provided by the DTE, leaving the MAU as simple as possible. This division of functions is based upon the recognition of the fact that since, in many cases, the MAU may be located in an inaccessible location adjacent to the physical medium, service of the MAU may often be difficult and expensive.

7.2 Functional Specification. The AUI is designed to make the differences among the various media as transparent as possible to the DTE. The selection of logical control signals and the functional procedures

are all designed to this end. Figure 7-2 is a reference model, a generalized MAU as seen by the DTE through the AUI.



NOTE: The AUI (comprised of DO, DI, CO, CI circuits) is not exposed when the MAU is, optionally, part of the DTE.

Fig 7-2
Generalized MAU Model

Many of the terms used in this section are specific to the interface between this sublayer and the MAC sublayer. These terms are defined in the Service Specification for the PLS sublayer.

7.2.1 PLS–PMA (DTE–MAU) Interface Protocol. The DTE and MAU communicate by means of a simple protocol across the AUI.

7.2.1.1 PLS to PMA Messages. The following messages can be sent by PLS sublayer entities in the DTE to PMA sublayer entities in the MAU:

Message	Meaning
output	Output information
output_idle	No data to be output
normal	Cease to isolate the MAU
(Optional)	
isolate	Isolate MAU
mau_request	Request that the MAU be made available

7.2.1.1.1 *output* Message. The PLS sublayer sends an output message to the PMA sublayer when the PLS sublayer receives an OUTPUT_UNIT from the MAC sublayer.

The physical realization of the *output* message is a CD0 or a CD1 sent by the DTE to the MAU on the Data Out circuit. The DTE sends a CD0 if the OUTPUT_UNIT is a ZERO or a CD1 if the OUTPUT_UNIT is a ONE. This message is time coded—that is, once this message has been sent, the function is not completed over the AUI until one bit time later. The *output* message cannot be sent again until the bit cell being sent as a result of sending the previous *output* message is complete.

7.2.1.1.2 *output_idle* Message. The PLS sublayer sends an *output_idle* message to the PMA sublayer at all times when the MAC sublayer is not in the process of transferring output data across the MAC to PLS interface. The *output_idle* message is no longer sent (and the first OUTPUT_UNIT is sent using the *output* message) as soon after the arrival of the first OUTPUT_UNIT as the MAU can be made available for data output. The *output_idle* message is again sent to the MAU when the DATA_COMPLETE is received from the MAC sublayer. The detailed usage of the *output_idle* message is shown in Fig 7-5.

The physical realization of the *output_idle* message is IDL sent by the DTE to the MAU on the Data Out circuit.

7.2.1.1.3 *normal* Message. The PLS sublayer sends a *normal* message to the PMA sublayer after it receives the PLS *start* message from the PLS Reset and Identify Function. The *normal* message is also sent after receipt of RESET_MONITOR_MODE from the management entity. The *normal* message is sent continuously by the PLS sublayer to the MAU, unless the PLS Output Function requires that the *mau_request* message be sent to permit data output. If *mau_request* is sent during data output, the sending of *normal* will be resumed when the PLS Output Function returns to the IDLE state. The *normal* signal is reset by the SET_MONITOR_MODE (this reset function is described more fully by Fig 7-4).

7.2.1.1.4 *isolate* Message (Optional). The PLS sublayer sends an *isolate* message to the PMA (in the MAU) whenever the PLS sublayer receives SET_MONITOR_MODE from the management entity. In response to the *isolate* message, the MAU causes the means employed to impress data on the physical medium to be positively prevented from affecting the medium. Since signaling and isolation techniques differ from medium to medium, the manner in which this positive isolation of the transmitting means is accomplished is specified in the appropriate MAU section. However, the intent of this positive isolation of the transmitter is to ensure that the MAU will not interfere with the physical medium in such a way as to affect transmissions of other stations even in the event that the means normally employed to prevent the transmitter from affecting the medium have failed to do so. The specification of positive isolation is not to be construed to preclude use of either active or passive devices to accomplish this function.

The physical realization of the *isolate* message is a CS0 signal sent by the DTE to the MAU over the Control Out circuit.

7.2.1.1.5 *mau_request* Message (Optional). The PLS sublayer sense the mau_request message to the PMA sublayer if the PMA sublayer is sending the *mau_not_available* message and the MAC sublayer has sent the first OUTPUT_UNIT of a new transmission. The PLS sublayer continues to send the *mau_request* message to the MAU until the MAC sublayer sends the DATA_COMPLETE request to the PLS sublayer across the MAC to PLS interface. See Figs 7-3, 7-5, and 7-9 for details.

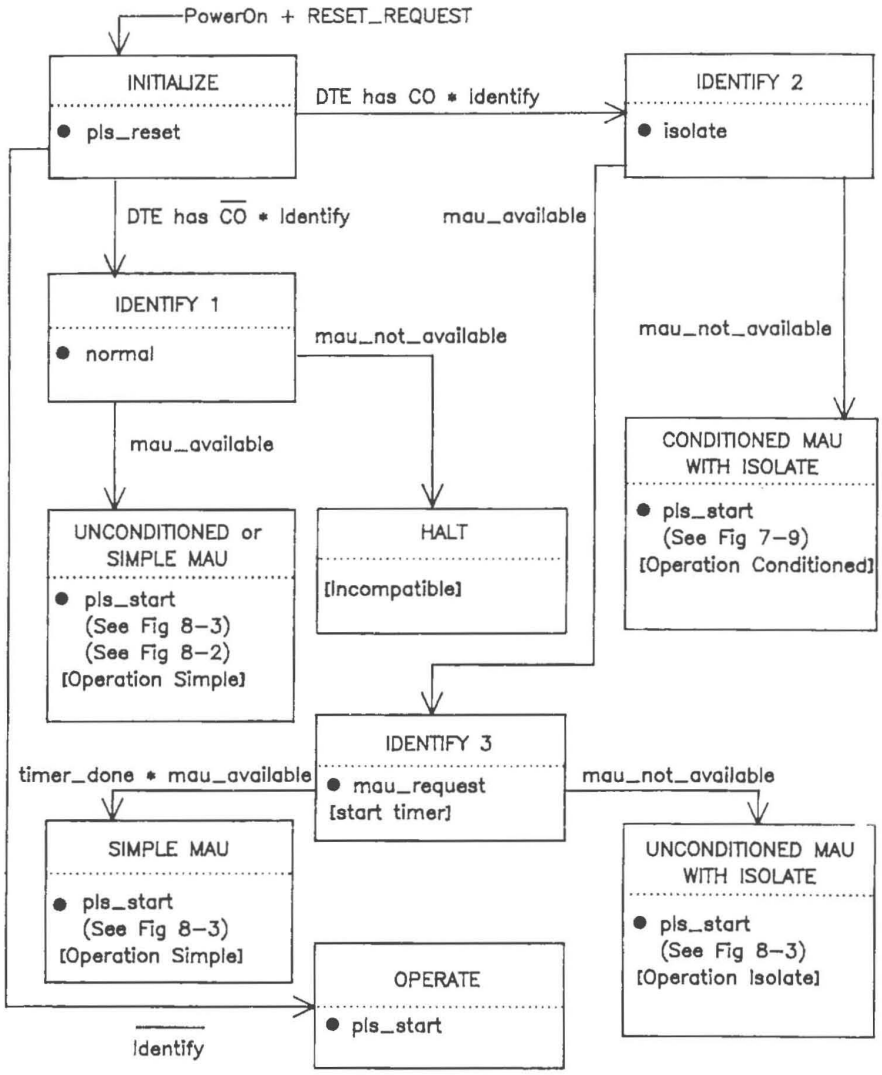
In addition, the *mau_request* message is used by the Reset and Identify Function in the IDENTIFY 3 state to determine whether the MAU has the Isolate Function.

The physical realization of *mau_request* is a CS1 sent by the DTE to the MAU on the Control Out circuit.

The physical realization of the *normal* message is the IDL signal sent by the DTE to the MAU on the Control Out circuit. In the absence of the CO circuit, MAUs implementing the Isolate Function shall act as if the *normal* message is present. The CO circuit components may be absent from the DTE, AUI, or MAU.

7.2.1.2 PMA to PLS Interface. The following messages can be sent by the Physical Medium Attachment sublayer entities in the MAU to the PLS sublayer entities in the DTE:

Message	Meaning
input	Input information
input_idle	No input information
signal_quality_error	Error detected by MAU
mau_available (Optional)	MAU is available for output
mau_not_available	MAU is not available for output



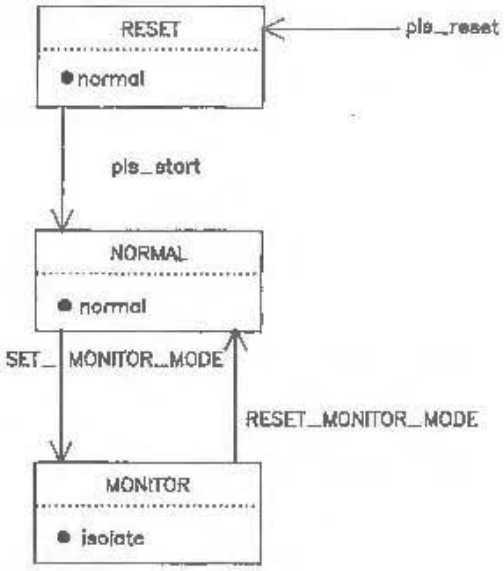
NOTES: (1) All states may be omitted except INITIALIZE and OPERATE
(2) "Identify" means DTE can recognize uniquely all CI messages and the entire function has been implemented
(3) "Identify" with bar means DTE fails to recognize *mau_not_available* or has a partial implementation of the function

Fig 7-3
PLS Reset and Identify Function

7.2.1.2.1 *input* Message. The PMA sublayer sends an *input* message to the PLS sublayer when the MAU has received a bit from the medium and is prepared to transfer this bit to the DTE. The actual mapping of the signals on the medium to the type of *input* message to be sent to the DTE is contained in the specifications for each specific MAU type. In general, when the *signal_quality_error* message is being sent by the MAU, the symmetry specifications for circuit DI are not guaranteed to be met.

The physical realization of the *input* message consists of CD0 or CD1 waveforms. If the *signal_quality_error* message is being sent from the MAU, the input waveform is unpredictable.

NOTE: This signal is not necessarily retimed by the MAU. Consult the appropriate MAU specification for timing and jitter.



NOTE: Monitor State is optional.

Fig 7-4
PLS Mode Function

7.2.1.2.2 *input_idle* Message. The PMA sublayer sends an *input_idle* message to the PLS sublayer when the MAU does not have data to send to the DTE.

The physical realization of the *input_idle* message is an IDL sent by the MAU to the DTE on the Data In circuit.

7.2.1.2.3 *signal_quality_error* Message. The PMA sublayer sends a *signal_quality_error* message to the PLS sublayer in response to any of three possible conditions. These conditions are improper signals on the medium, collision on the medium, and reception of the *output_idle* message. They are described in the following numbered paragraphs. The physical realization of the *signal_quality_error* message is a CS0 sent by the MAU to the DTE on the Control In circuit.

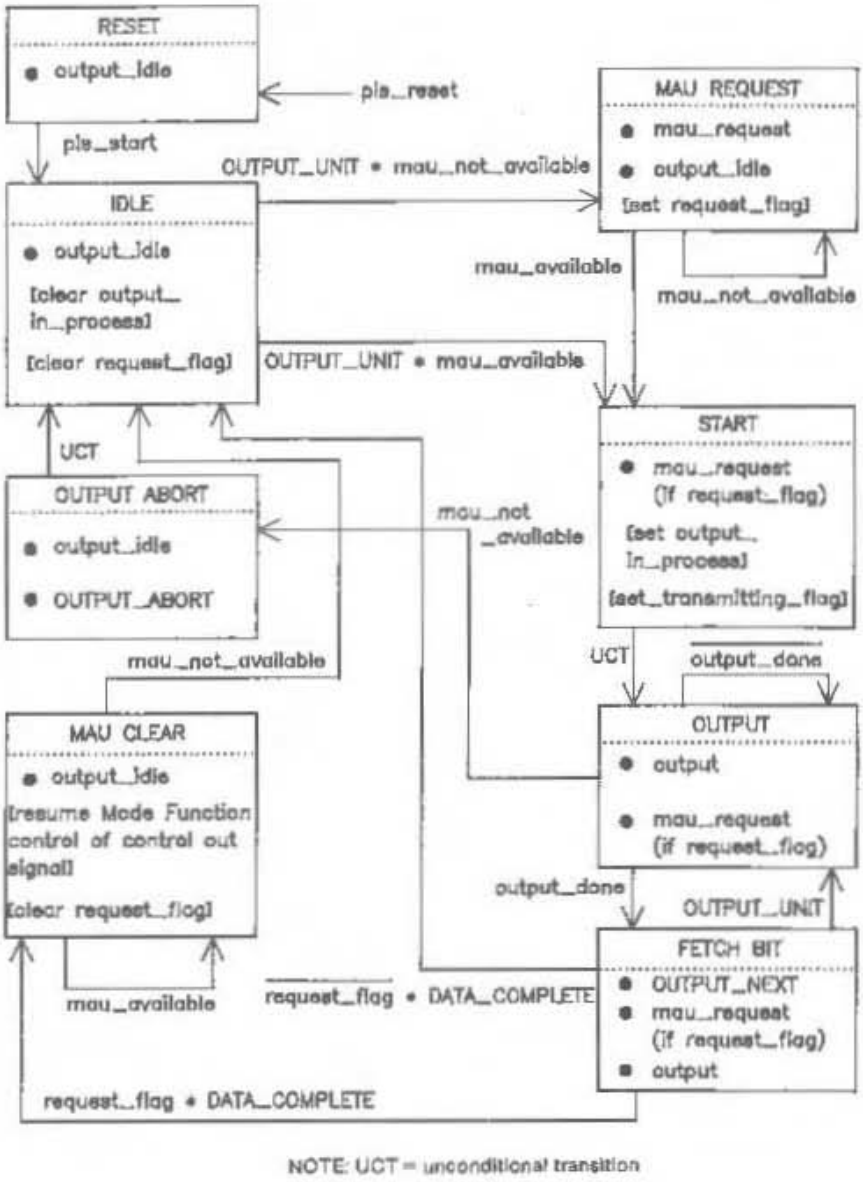
NOTE: The MAU is required to assert the *signal_quality_error* message at the appropriate times whenever the MAU is powered, and not just when the DTE is requesting data output. See Figs 7-9, 8-2, and 8-3 for details.

(1) Improper Signals on the Medium. The MAU may send the *signal_quality_error* message at any time due to improper signals on the medium. The exact nature of these improper signals are medium-dependent. Typically, this condition might be caused by a malfunctioning MAU (for example, repeater or head-end) connected to the medium or by a break or short in the medium. See the appropriate MAU specification for specific conditions that may cause improper signals on a given medium.

(2) Collision. Collision occurs when more than one MAU is transmitting on the medium. The local MAU shall send the *signal_quality_error* message in every instance when it is possible for it to ascertain that more than one MAU is transmitting on the medium. The MAU shall make the best determination possible. The MAU shall not send the *signal_quality_error* message when it is unable to determine conclusively that more than one MAU is transmitting.

(3) *signal_quality_error* Message Test. The MAU sends the *signal_ quality_error* message at the completion of the Output Function. See Fig 7-9 and Section 8 for a more complete description of this test.

7.2.1.2.4 *mau_available* Message. The PMA sublayer sends the *mau_available* message to the PLS sublayer when the MAU is available for output. The *mau_available* message is always sent by a MAU that is always prepared to output data except when it is required to signal the *signal_quality_error*



NOTE: UCT = unconditional transition

Fig 7-5
PLS Output Function

message. Such a MAU does not require *mau_request* to prepare itself for data output. See Figs 7-3, 7-5, and 7-9 for details.

The physical realization of the *mau_available* message is an IDL sent by the MAU to the DTE on the Control In circuit.

7.2.1.2.5 *mau_not_available* Message (OPTIONAL). The PMA sublayer sends a *mau_not_available* message to the PLS sublayer when the MAU is not available for output. Figure 7-5 shows the relationship of *mau_not_available* to the Output Function.

The *mau_not_available* message is also used by a MAU that contains the Isolate Function and does not need to be conditioned for output to signal the presence of the Isolate Function during the PLS Reset Function (see Fig 7-3 and 8-3).

The physical realization of the *mau_not_available* message is a CS1 sent by the MAU to the DTE on the Control In circuit.

7.2.2 PLS Interface to MAC and Management Entities. The PLS sublayer interfaces described here are for reference only. This section specifies the services sent between the MAC sublayer and the PLS sublayer.

7.2.2.1 PLS–MAC Interface. The following messages can be sent between PLS sublayer entities and MAC sublayer entities:

Message	Meaning
OUTPUT_UNIT	Data sent to the MAU
OUTPUT_STATUS	Response to OUTPUT_UNIT
INPUT_UNIT	Data received from the MAU
CARRIER_STATUS	Indication of input activity
SIGNAL_STATUS	Indication of error/no error condition

7.2.2.1.1 OUTPUT_UNIT. The MAC sublayer sends the PLS sublayer an OUTPUT_UNIT every time the MAC sublayer has a bit to send. Once the MAC sublayer has sent an OUTPUT_UNIT to the PLS sublayer, it may not send another OUTPUT_UNIT until it has received an OUTPUT_STATUS message from the PLS sublayer. The OUTPUT_UNIT is a ONE if the MAC sublayer wants the PLS sublayer to send a CD1 to the PMA sublayer, a ZERO if a CD0 is desired, or a DATA_COMPLETE if an IDL is desired.

7.2.2.1.2 OUTPUT_STATUS. The PLS sublayer sends the MAC sublayer OUTPUT_STATUS in response to every OUTPUT_UNIT received by the PLS sublayer. OUTPUT_STATUS sent is an OUTPUT_NEXT if the PLS sublayer is ready to accept the next OUTPUT_UNIT from the MAC sublayer, or an OUTPUT_ABORT if the PLS sublayer was not able to process the previous OUTPUT_UNIT. (The purpose of OUTPUT_STATUS is to synchronize the MAC sublayer data output with the data rate of the physical medium.)

7.2.2.1.3 INPUT_UNIT. The PLS Sublayer sends the MAC sublayer an INPUT_UNIT every time the PLS receives an *input* message from the PMA sublayer. The INPUT_UNIT is a ONE if the PLS sublayer receives a CD1 from the PMA sublayer, a ZERO if the PLS sublayer receives a CD0 from the PMA sublayer.

7.2.2.1.4 CARRIER_STATUS. The PLS sublayer sends the MAC sublayer CARRIER_STATUS whenever the PLS sublayer detects a change in carrier status. The PLS sublayer sends CARRIER_ON when it receives an *input* or *signal_quality_error* message from the PMA and the previous CARRIER_STATUS that the PLS sublayer sent to the MAC sublayer was CARRIER_OFF. The PLS sublayer sends CARRIER_OFF when it receives an *input_idle* from the PMA sublayer, no *signal_quality_error* (either *mau_available* or *mau_not_available*) message and the previous CARRIER_STATUS that the PLS sublayer sent to the MAC sublayer was CARRIER_ON.

7.2.2.1.5 SIGNAL_STATUS. The PLS sublayer sends the MAC sublayer SIGNAL_STATUS whenever the PLS sublayer detects a change in the signal quality (as reported by the PMA). The PLS sublayer sends SIGNAL_ERROR when it receives a *signal_quality_error* message from the PMA sublayer and the previous SIGNAL_STATUS the PLS sublayer sent was NO_SIGNAL_ERROR. The PLS sublayer sends NO_SIGNAL_ERROR when it receives no *signal_quality_error* (either *mau_available* or *mau_not_available)* message from the PMA sublayer and the previous CARRIER_STATUS that the PLS sent to the MAC sublayer was SIGNAL_ERROR.

7.2.2.2 PLS-Management Entity Interface. The following messages may be sent between the PLS sublayer entities and intralayer or higher layer management entities:

Message	Meaning
RESET_REQUEST	Reset PLS to initial "Power On" state
RESET RESPONSE	Provides operational information
MODE_CONTROL	Control operation
SQE_TEST	Signal Quality Error test results

7.2.2.2.1 RESET_REQUEST. The management entity sends the PLS sublayer RESET_REQUEST when the PLS sublayer needs to be reset to a known state. Upon receipt of RESET_REQUEST, the PLS sublayer resets all internal logic and restarts all functions. See Fig 7-3 for details.

7.2.2.2.2 RESET_RESPONSE. The PLS sublayer sends the management entity RESET_R SPONSE upon completion of the Reset and Identify Function (see Fig 7-3 and 7.2.4.1) whether invoked due to power on or due to a RESET_REQUEST. Which RESET_RESPONSE was sent is deter mined by the Reset and Identify Function. A RESET_RESPONSE of OPERATION SIMPLE, OPERATION ISOLATE, or OPERATION CONDITIONED is sent if the MAU is compatible with the DTE and the MAU is simple (no isolate) or if the DTE does not support Isolate even if Isolate is supported by the MAU, supports Isolate but does not require conditioning, or supports Isolate and does require conditioning to output. A RESET_RESPONSE of INCOMPATIBLE is sent if the MAU is not compatible with the DTE (that is, the MAU requires conditioning but the DTE does not support conditioning).

7.2.2.2.3 MODE_CONTROL. The management entity sends MODE_CONTROL to the PLS sublayer to control PLS functions. MODE_CONTROL capabilities are as follows:

Message	Meaning
ACTIVATE PHYSICAL	Supply power on circuit VP
DEACTIVATE PHYSICAL	Remove power from circuit VP
SET_MONITOR_MODE	Send Isolate to MAU
RESET_MONITOR_MODE	Send Normal to MAU

7.2.2.2.4 SQE_TEST. The PLS sublayer sends SQE_TEST to the management entity at the conclusion of each *signal_quality_error* test (see Output Function, 7.2.4.3). The PLS sublayer sends SQE_TEST_ERROR if the *signal_quality_error* test fails or SQE_TEST_OK if the *signal_quality_error* test passes.

7.2.3 Frame Structure. Frames transmitted on the AUI shall have the following structure:

<silence><preamble><sfd><data><etd><silence>

The frame elements shall have the following characteristics:

Element	Characteristics
<silence>	= no transitions
<preamble>	= alternating (CD1) and (CD0) | 56 bit times (ending in CD0)
<sfd>	= (CD1)(CD0)(CD1)(CD0)(CD1)(CD0)(CD1)(CD1)
<data>	= 8 × N
<etd>	= IDL

7.2.3.1 Silence. The <silence> delimiter provides an observation window for an unspecified period of time during which no transitions occur on the AUI. The minimum length of this period is specified by the access procedure.

7.2.3.2 Preamble. The <preamble> delimiter begins a frame transmission and provides a signal for receiver synchronization. The signal shall be an alternating pattern of (CD1) and (CD0). This pattern shall be transmitted on the Data Out circuit by the DTE to the MAU for a minimum of 56 bit times at the beginning of each frame. The last bit of the preamble (that is, the final bit of preamble before the start of frame delimiter) shall be a CD0.

The DTE is required to supply at least 56 bits of preamble in order to satisfy system requirements. System components consume preamble bits in order to perform their functions. The number of preamble bits sourced ensures an adequate number of bits are provided to each system component to correctly implement its function.

7.2.3.3 Start of Frame Delimiter (SFD). The <sfd> indicates the start of a frame, and follows the preamble. The <sfd> element of a frame shall be

(CD1)(CD0)(CD1)(CD0)(CD1)(CD0)(CD1)(CD1)

7.2.3.4 Data. The <data> in a transmission shall be in multiples of eight (8) encoded data bits (CD0s and CD1s).

7.2.3.5 End of Transmission Delimiter. The <etd> delimiter indicates the end of a transmission and serves to turn off the transmitter. The signal shall be an IDL.

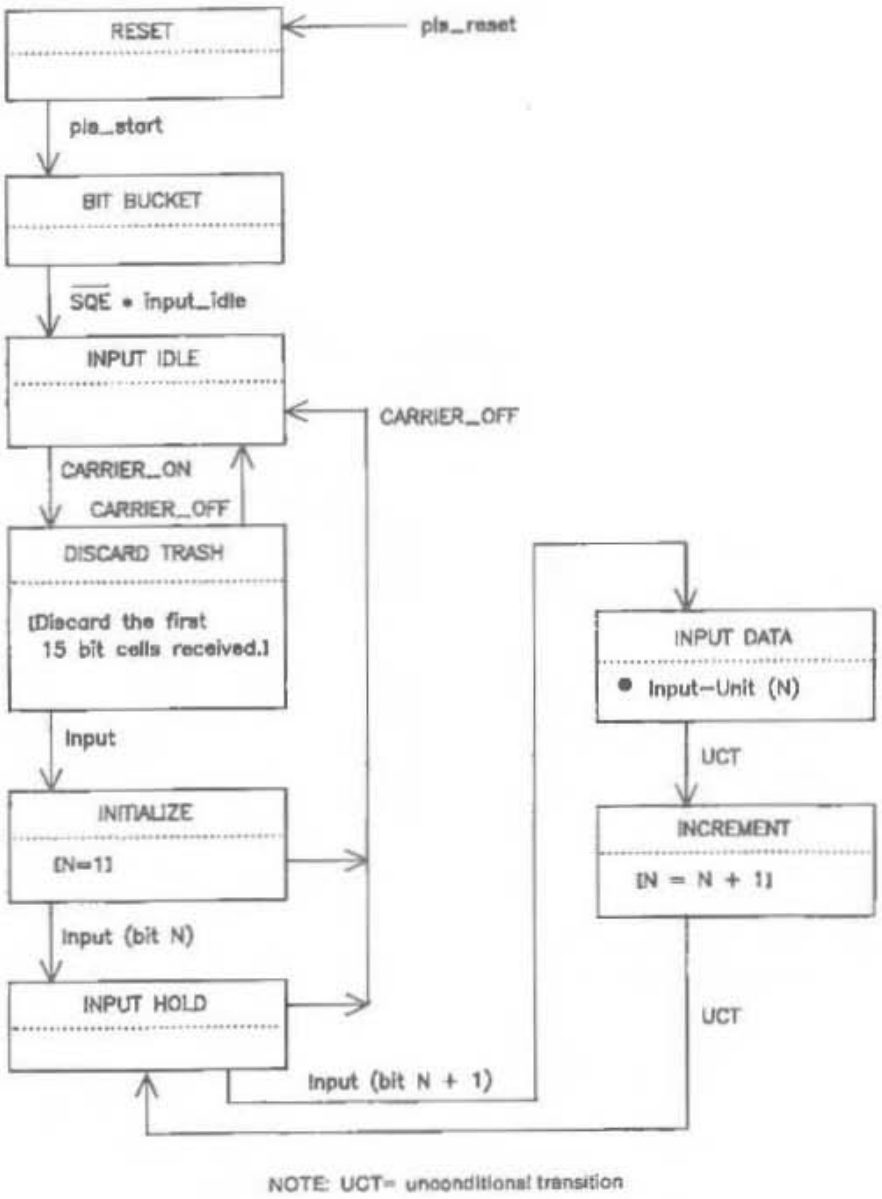
7.2.4 PLS Functions. The PLS sublayer functions consist of a Reset and Identify Function and five simultaneous and asynchronous functions. These functions are Output, Input, Mode, Error Sense, and Carrier Sense. All of the five functions are started immediately following the completion of the Reset and Identify Function. These functions are depicted in the state diagrams shown in Figs 7-3 through 7-8, using notation described in 1.2.1.

7.2.4.1 Reset and Identify Function. The Reset and Identify Function is executed any time either of two conditions occur. These two conditions are "power on" and the receipt of RESET_REQUEST from the management entity. The Reset and Identify Function initializes all PLS functions, and (optionally) determines the capability of the MAU attached to the AUI. Figure 7-3 is the state diagram of the Reset and Identify Function. The Identify portion of the function is optional.

7.2.4.2 Mode Function. The MAU functions in two modes: normal and monitor. The monitor mode is optional. The state diagram of Fig 7-4 depicts the operation of the Mode Function. When the MAU is operating in the normal mode, it functions as a direct connection between the DTE and the medium. Data sent from the DTE are impressed onto the medium by the MAU and all data appearing on the medium are sent to the DTE by the MAU. When the MAU is operating in the monitor mode, data appearing on the medium is sent to the DTE by the MAU as during the normal mode. *signal_quality_error* is also asserted on the AUI as during operation in the normal mode. However, in the monitor mode, the means employed to impress data on the physical medium is positively prevented from affecting the medium. Since signaling and isolation techniques differ from medium to medium, the manner in which this positive isolation of the transmitting means is accomplished is specified in the appropriate MAU document. However, the intent of this positive isolation of the transmitter is to ensure that the MAU will not interfere with the physical medium in such a way as to affect transmission of other stations even in the event of failure of the normal transmitter disabling control paths within the transmitting mechanism of the MAU.

The monitor mode is intended to permit a network station to determine if it is the source of interference observed on the medium.

NOTE: The monitor mode is intended to be used only by Network Management for fault isolation and network operation verification. It is intended that the *isolate* message provide direct control over the mode function so that these tasks can be performed. IMPROPER USE OF THE ISOLATE FUNCTION CAN CAUSE ERRONEOUS FRAMES. Section 5, Layer Management, provides details on the proper use of this function.

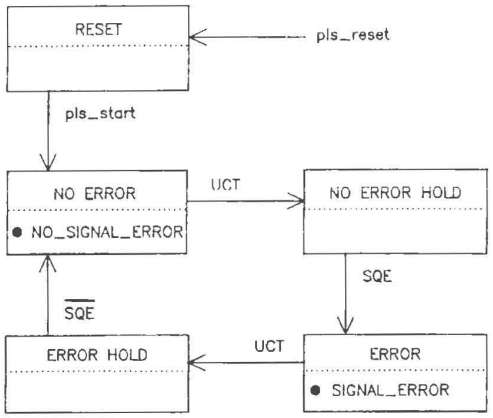


NOTE: UCT= unconditional transition

Fig 7-6
PLS Input Function

7.2.4.3 Output Function. The PLS sublayer Output Function transparently performs the tasks of conditioning the MAU for output and data transfer from the MAC sublayer to the MAU. The state diagram of Fig 7-5 depicts the Output Function operation.

At the conclusion of the Output Function, if a collision has not occurred, a test is performed to verify operation of the signal quality detection mechanism in the MAU and to verify the ability of the AUI to pass the *signal_quality_error* message to the PLS sublayer. The operation of this test in the DTE is shown in Fig 7-8.



NOTE: UCT = unconditional transition

Fig 7-7
PLS Error Sense Function

7.2.4.4 Input Function. The PLS sublayer Input Function transparently performs the task of data transfer from the MAU to the MAC sublayer. The state diagram of Fig 7-6 depicts the Input Function operation.

7.2.4.5 Error Sense Function. The PLS sublayer Error Sense Function performs the task of sending SIGNAL_STATUS to the MAC sublayer whenever there is a change in the signal quality information received from the MAU. The state diagram of Fig 7-7 depicts the Error Sense Function operation.

7.2.4.6 Carrier Sense Function. The PLS sublayer Carrier Sense Function performs the task of sending CARRIER_STATUS to the MAC sublayer every time there is a change in CARRIER_STATUS. The state diagram of Fig 7-8 depicts the Carrier Sense Function operation.

Verification of the *signal_quality_error* detection mechanism occurs in the following manner (in the absence of a fault on the medium).

(1) At the conclusion of the output function, the DTE opens a time window during which it expects to see the *signal_quality_error* signal asserted on the Control In circuit. The time window begins when CARRIER_STATUS becomes CARRIER_OFF. If execution of the Output Function does not cause CARRIER_ON to occur, no SQE test occurs in the DTE. The duration of the window shall be at least 4.0 μs but no more than 8.0 μs. During the time window (depicted as carrier_inhibit_timer, Fig 7-8) the Carrier Sense Function is inhibited.

(2) The MAU, upon waiting Tw (wait time) after the conclusion of output, activates as much of the signal quality error detecting mechanism as is possible without placing signals on the medium, thus sending the *signal_quality_error* message across the AUI for 10 ± 5 bit times (10/BR ± 5/BR seconds).

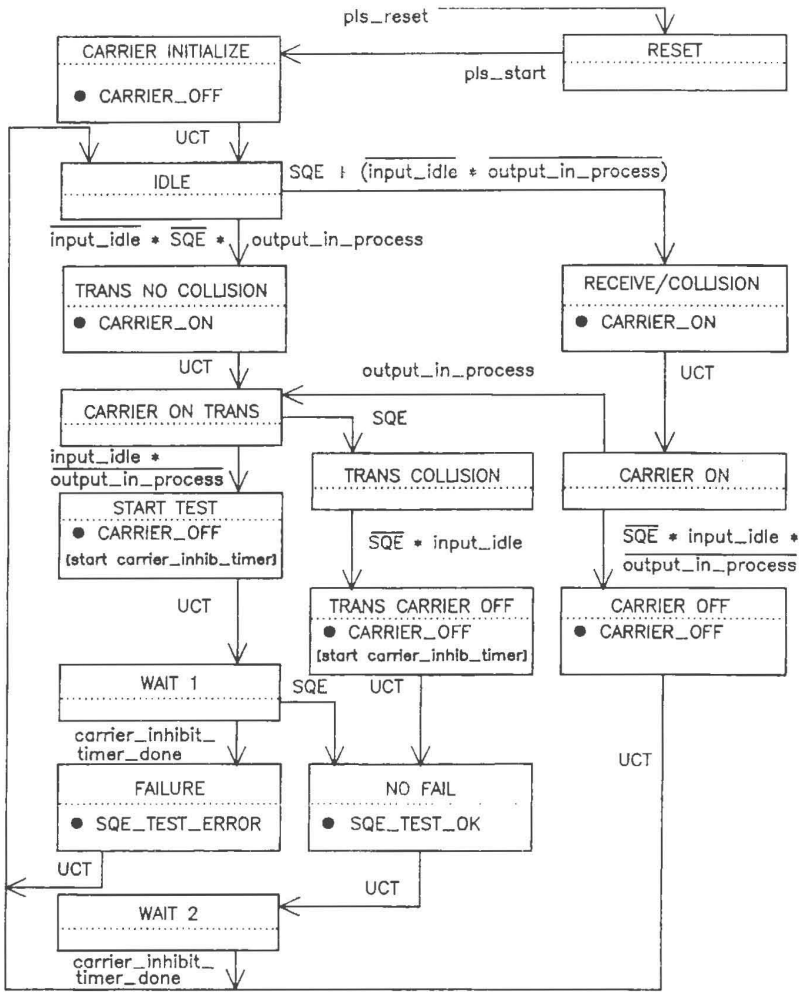
(3) The DTE interprets the reception of the *signal_quality_error* message from the MAU as indication that the *signal_quality_error* detecting mechanism is operational and the *signal_quality_error* message may be both sent by the MAU and received by the DTE.

NOTES: (1) The occurrence of multiple (overlapping) transmitters on the medium during the time that the test window is open, as specified above, will satisfy the test and will verify proper operation of the signal quality error detecting mechanism and sending and receiving of the appropriate physical error message.

(2) If *signal_quality_error* exists at the DTE before CARRIER_OFF occurs, then the Collision Presence test sequence within the PLS as described in 7.2.4.3 above shall be aborted as shown in Fig 7-8.

7.3 Signal Characteristics

7.3.1 Signal Encoding. Two different signal encoding mechanisms may be used by the AUI. One of the mechanisms is used to encode data, the other to encode control.

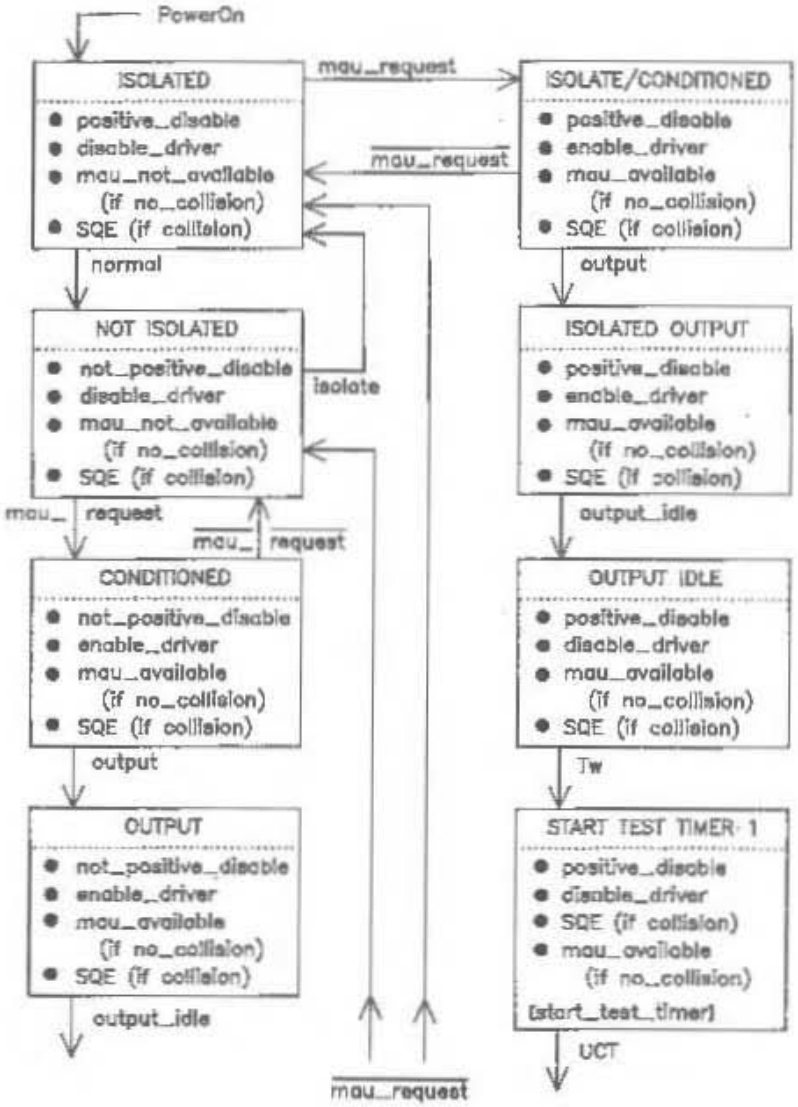


NOTE: UCT = unconditional transition
SQE = *signal_quality_error*

Fig 7-8
PLS Carrier Sense Function

7.3.1.1 Data Encoding. Manchester encoding is used for the transmission of data across the AUI. Manchester encoding is a binary signaling mechanism that combines data and clock into "bit-symbols." Each bit-symbol is split into two halves with the second half containing the binary inverse of the first half; a transition always occurs in the middle of each bit-symbol. During the first half of the bit-symbol, the encoded signal is the logical complement of the bit value being encoded. During the second half of the bit-symbol, the encoded signal is the uncomplemented value of the bit being encoded. Thus, a CD0 is encoded as a bit-symbol in which the first half is HI and the second half is LO. A CD1 is encoded as a bit-symbol in which the first half is LO and the second half is HI. Examples of Manchester waveforms are shown in Fig 7-10.

The line condition IDL is also used as an encoded signal. An IDL always starts with a HI signal level. Since IDL always starts with a HI signal, an additional transition will be added to the data stream if the last bit sent was a zero. This transition cannot be confused with clocked data (CD0 or CD1) since the transition will occur at the start of a bit cell. There will be no transition in the middle of the bit cell. The IDL



(a)

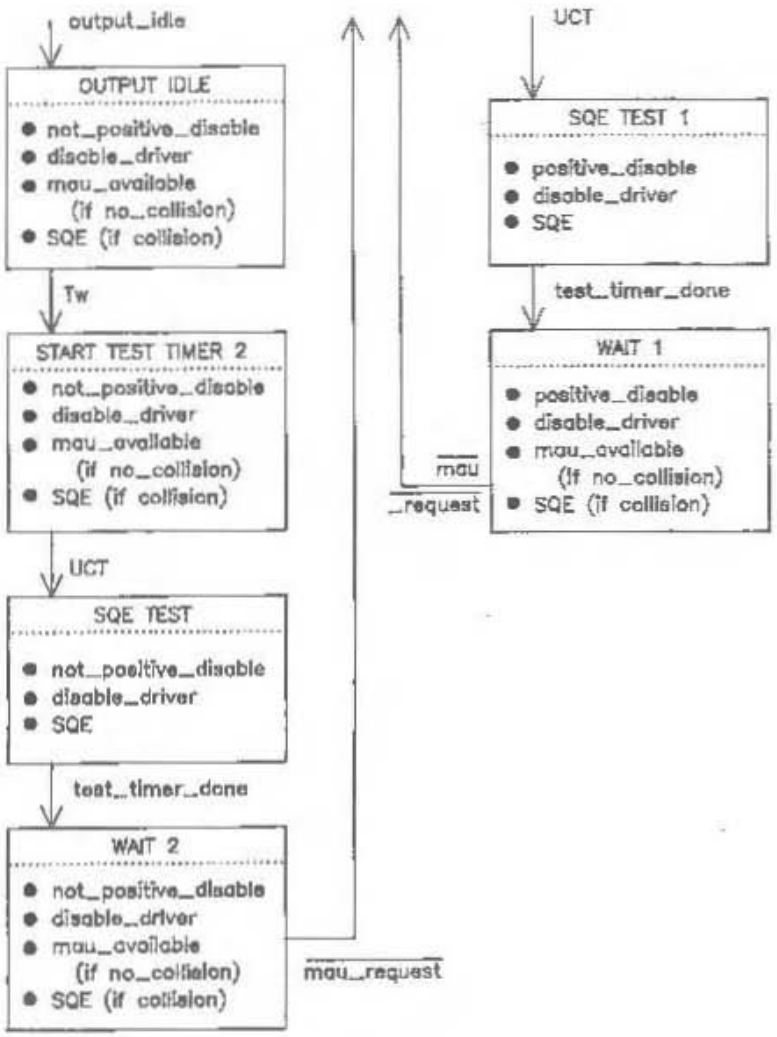
NOTE: See Figs 8-2 and 8-3 for simple and isolate type MAUs.

Fig 7-9
Interface Function for MAU with Conditioning

condition, as sent by a driver, shall be maintained for a minimum of 2 bit times. The IDL condition shall be detected within 1.6 bit times at the receiving device.

(1) System jitter considerations make detection of IDL (etd, end transmission delimiter) earlier than 1.3 bit times impractical. The specific implementation of the phase-locked loop or equivalent clock recovery mechanism determines the lower bound on the actual IDL detection time. Adequate margin between lower bound and 1.6 bit times should be considered.

(2) Recovery of timing implicit in the data is easily accomplished at the receiving side of the interface because of the wealth of binary transitions guaranteed to be in the encoded waveform, independent



(b)

NOTE: See Figs 8-2 and 8-3 for simple and isolate type MAUs.

Fig 7-9
Interface Function for MAU with Conditioning

of the data sequence. A phase-locked loop or equivalent mechanism maintains continuous tracking of the phase of the information on the Data circuit.

7.3.1.2 Control Encoding. A simpler encoding mechanism is used for control signaling than for data signaling. The encoded symbols used in this signaling mechanism are CS0, CS1, and IDL. The CS0 signal is a signal stream of frequency equal to the bit rate (BR). The CS1 signal is a signal stream of frequency equal to half of the bit rate (BR/2). If the interface supports more then one bit rate (see 4.2), the bit rate in use on the data circuits is the one to which the control signals are referenced. The IDL signal used on the control circuits is the same as the IDL signal defined for the data circuits (see 7.3.1.1). The Control Out circuit is optional (O) as is one message on Control In.

The frequency tolerance of the CS1 and CS0 signals on the CO circuit shall be ±5% and that of the CS1 signal on the CI circuit shall be ±15%. The duty cycle of the above signals is nominally 50%/50% and shall

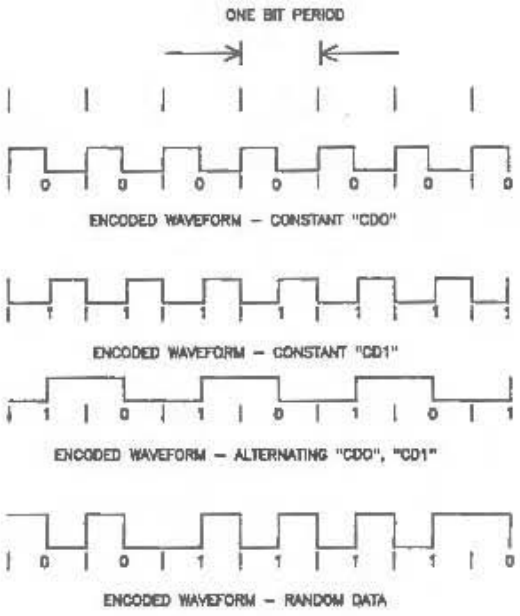


Fig 7-10
Examples of Manchester Waveforms

be no worse than 60%/40%. The CS0 signal on the CI circuit shall have a frequency tolerance of BR +25%, –15% with the pulse widths no less than 35 ns and no greater than 70 ns at the zero crossing points.

The meaning of the signals on the Control Out circuit (DTE to MAU) are:

Signal	Message	Description
IDL	*normal*	Instructs the MAU to enter (remain in) normal mode
CS1	*mau_request* (O)	Requests that the MAU should be made available
CS0	*isolate* (O)	Instructs the MAU to enter (remain in) monitor mode

The meaning of the signals on the Control In circuit (MAU to DTE) are:

Signal	Message	Description
IDL	*mau_available*	Indicates that the MAUs ready to output data
CS1	*mau_not_available*	Indicates that the MAU is not ready to output data
CS0	*signal_quality_error*	Indicates that the MAU has detected an error output data

7.3.2 Signaling Rate. Signaling rates of from 1 to 20 Mb/s are encompassed by this standard. This edition of the standard specifies a signaling rate of 10 million bits per second ± 0.01%.

It is intended that a given MDI operate at a single data rate. It is not precluded that specific DTE and MAU designs be manually switched or set to alternate rates. A given local network shall operate at a single signaling rate. To facilitate the configuration of operational systems, DTE and MAU devices shall be labeled with the actual signaling rate used with that device.

7.3.3 Signaling Levels. Exact voltage and current specifications are listed in 7.4.