DIGITAL IMAGING & COMMUNICATIONS IN MEDICINE (DICOM)

PART 9

Point to Point Communication Support for Message Exchange

PS 3.9-1993

Digital Imaging and Communications in Medicine (DICOM)
Part 9: Point to Point Communication Support for Message Exchange

Approved by:

The American College of Radiology

Published by:

National Electrical Manufacturers Association
1300 North 17th Street, Suite 1847
Rosslyn, Virginia 22209

CONTENTS

ACR-NEMA

Digital Imaging and Communications in Medicine (DICOM)
PART 9: Point to Point Communication Support for Message Exchange

FOREWORD

ACR (the American College of Radiology) and NEMA (the National Electrical Manufacturers Association) formed a joint committee to develop a Standard for Digital Imaging and Communications in Medicine. This DICOM Standard was developed according to the NEMA Procedures.

This Standard is developed in liaison with other Standard Organizations including CEN TC251 in Europe and JIRA in Japan, with review also by other organizations including IEEE, HL7 and ANSI in the USA.

The DICOM Standard is structured as a multi-part document using the guidelines established in the following document:

— IEC/ISO Directives, 1989 Part 3—Drafting an presentation of International Standards.

This document is one of the Parts of the DICOM Standard which consists of the following Parts:

PS 3.1: Introduction and Overview

PS 3.2: Conformance

PS 3.3: Information Object Definitions

PS 3.4: Service Class Specifications

PS 3.5: Data Structure and Semantics

PS 3.6: Data Dictionary

PS 3.7: Message Exchange Protocol

PS 3.8: Network Communications Support for Message Exchange

PS 3.9: DICOM Point-to-Point Communications Support for Message Exchange

Digital Imaging and Communications in Medicine
PART 9: Point to Point Communication Support for Message Exchange

1 Scope and field of application

This part of the DICOM Standard specifies the services and protocols necessary to support the DICOM Application Entity over a point to point interface. This part shall be used in conjunction with other Parts as presented in Part 1 - Introduction and Overview.

The interface specified in this Part is backwards compatible with the interface specified in ACR-NEMA 300-1988.

The point to point Session/Transport/Network Layer interface and services are largely compatible with the Upper Layer interface and services described in Part 8 of the DICOM Standard. This compatibility facilitates the development of applications intended for use in either the DICOM point to point or network environment.

The communication protocol defined in this Part of the Standard makes use of the ISO/OSI reference model for communication protocols. This model defines a seven layer approach to sending and receiving information between two communicating entities. The point to point interface defined here specifies the physical, datalink, network, transport and session layer functionality needed. The three Layers defined in this Part of the Standard that accomplish this functionality are called as the Physical, Data Link and the Session/Transport/Network Layers.

The Physical Layer protocol handles control signal handshakes and interrupts and the transfer of data using the electrical lines of the interface. It is defined by a description, a state diagram, and a timing diagram.

Data flow is supported by the use of a frame protocol at the Data Link Layer.
A packet protocol at the Session/Transport/Network Layer supports virtual channels and conveys segments of a message across the interface. End to end message delivery is also supported by the Session/Transport/Network Layer.

Required peer-to-peer formats and transactions are specified here, but the implementation of inter-layer communications internal to each device is designed by the vendor. The specified protocols support the transmission of DICOM messages from the Application Layer with virtual connections at the Session/Transport/Network Layer. These connections are opened and closed with the specified services to support communication. A message is fragmented into blocks of up to 2048 16-bit words and combined with the channel identification number and a block sequence number to make data packets. The packets are delivered to the Data Link Layer where they are framed by a frame descriptor word and frame check sequence. The Data Link Layer also handles interface status and error reporting. The frames are then sent to the Physical Layer for transmission across the physical interface. The Physical Layer handles the hardware interface arbitration and data word transfer protocol.

Each frame of a message has the form shown in Figure 1-1, where the components of a frame, packet and message fragment are illustrated.

The point to point interface plane across which the data moves is located between the imaging equipment and a second device, as shown in Figure 1-2. In more complex systems, the second device could be a network interface unit to provide access to an interconnecting network. The DICOM point to point interface standard imposes no specification on such external networks.

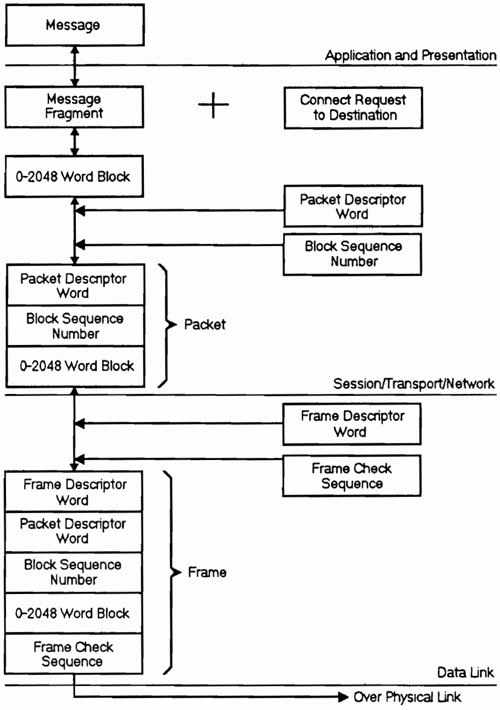


Figure 1-1—DICOM point to point message handling

Figure 1-2—Location of DICOM point to point

2 Normative references

2.1 *International standards*

ISO 7498, Information Processing Systems - Open Systems Interconnection - Basic Reference Model.
ISO/TR 8509 - Information Processing Systems - Open Systems Interconnection - Service Conventions
2.2 Other Documents EIA 485, Standard for Electrical Characteristics of Generators and Receivers for Use in Balanced Digital Multipoint Systems.

3 Definitions

3.1 *Reference model definitions*

This part of the Standard is based on the concepts developed in ISO 7498 and makes use of the following terms defined in it:

a) Application Entity
b) Application Layer
c) Data Link Layer
d) Physical Layer
e) Presentation Layer
f) Protocol
g) Protocol Data Unit (PDU)
h) Service
i) Session Layer

3.2 *Service convention definitions*

This part of the Standard makes use of the following terms or concepts defined in ISO/TR 8509:

a) Service Provider
b) Service User
c) Confirmed Service
d) Non-confirmed Service
e) Primitive
f) Request (Primitive)
g) Indication (Primitive)
h) Response (Primitive)
i) Confirmation (Primitive)

3.3 *DICOM network communication support definitions*

This part of the Standard makes use of the following terms defined in Part 8 of the Standard:

a) Upper Layer
b) Network Interface Unit (NIU)

3.4 *DICOM point to point communication support definitions*

The following definitions are commonly used in this Part of the Standard:

3.4.1 frame: A Data Link data unit containing a Frame Descriptor Word followed by a Sessions/Transport/Network Layer packet and a Frame Check Sequence. These data units are exchanged between peer Data Link entities.

3.4.2 packet: A Session/Transport/Network data unit containing a Packet Header followed by a Connection Header and then a block of message data. These data units are exchanged between peer Session/Transport/Network Layers.

3.4.3 session/transport/network layer: The session, transport and network layers have been merged together in this Part of the Standard. There functionality is defined by the single layer called the Session/Transport/Network (STN) Layer.

4 Symbols and abbreviations

4.1 **ACR** American College of Radiology

4.2 **AE** Application Entity

4.3 **BSN** Block Sequence Number

4.4 **CCI** Close Channel Indication

4.5 **CCR** Close Channel Request

4.6 **CMD** Command

4.7 **DICOM** Digital Imaging and Communications in Medicine

4.8 **EIA** Electronic Industries Association

4.9 **INTI** Interrupt In

4.10 **INTO** Interrupt Out

4.11 **ISO** International Standards Organization

4.12 **NEMA** National Electrical Manufacturers Association

4.13 **NIU** Network Interface Unit

4.14 **OCI** Open Channel Indication

4.15 **OCR** Open Channel Request

4.16 **OSI** Open Systems Interconnection

4.17 **PAR** Parity

4.18 **PDU** Protocol Data Unit

4.19 **REQI** Request to Send In

4.20 **REQO** Request to Send Out

4.21 **STBI** Strobe In

4.22 **STBO** Strobe Out

4.23 **STN** Session/Transport/Network

5 Conventions

The following conventions are used for the service description tables shown in this Part of the Standard.

(=) The parameter value used in the indication/confirmation shall be the same as used in the request/response.

A- A prefix used in service names and primitives that indicate the Application layer is making use of the service or primitive.

M Mandatory usage.

MF Mandatory with a fixed value.

N- A prefix used in service names and primitives that indicate the network layer is making use of the service or primitive.

P Provider initiated

P- A prefix used in service names and primitives that indicate the presentation layer is making use of the service or primitive.

S- A prefix used in service names and primitives that indicate the session layer is making use of the service or primitive.

T- A prefix used in service names and primitives that indicate the transport layer is making use of the service or primitive.

6 Point to point session/transport/network layer protocol

Medical imaging equipment that uses the point to point interface may be performing multiple tasks and an external network may also be involved. Therefore, a protocol is needed to establish end-to-end message connections using Session/Transport/Network (STN) Layer virtual channels and to transmit and receive DICOM messages. The protocol defined in this section manages the connection, fragments and rebuilds DICOM messages and sends/receives these packets via the Data Link Layer. A STN Layer State Machine is specified in terms of STN Layer user services, abstract primitives and Protocol Data Units (PDUs).

The peer to peer protocol specified here uses numbered packets to support the fragmentation and reconstruction of DICOM messages. Virtual channels allow for the existence of multiple simultaneous associations. Each association supports only one channel.

Only one transfer syntax shall be supported for point to point message exchange: little endian. See Part 5 of this Standard for transfer syntax encoding definitions.

Note— The terms connection and association are used interchangeably in this section.

Four services are provided by the STN Layer: association establishment (connection), association release (disconnection), provider initiated abort and data transfer. These services and their types are listed in Table 6-1 and further discussed in Part 8 of this Standard.

Table 6-1—Session/transport/network services

Service	Type
A-ASSOCIATE	Confirmed
A-RELEASE	Confirmed
A-P-ABORT	Provider-initiated
P-DATA	Non-Confirmed

The STN Layer services are provided via 11 primitives. These primitives are passed between the STN Layer and it's users, ie., any Application or Presentation Entity. These primitives are required in order for the STN Layer to accomplish the services described in the following sub-sections.

6.1 *Association service*

The STN Layer Association service is a confirmed service that supports the following four primitives:

a) A-ASSOCIATE Request

This primitive is passed from the requesting Application Entity to its STN Layer in order to request that a new channel be opened and an association be established with that channel.

b) A-ASSOCIATE Indication

This primitive is passed from the STN Layer to the accepting Application Entity in order to inform the application that a peer wishes to open a channel and establish an association.

c) A-ASSOCIATE Response

This primitive is passed from the accepting Application Entity to its STN Layer in order to accept the opening of a channel and the establishment of an association requested by a peer.

d) A-ASSOCIATE Confirmation

This primitive is passed from the STN Layer to the requesting Application Entity in order to inform the application that a peer has opened a channel and accepted the association.

6.1.1 A-ASSOCIATE PARAMETERS

Table 6.1-1 lists the parameters that are required for an A-ASSOCIATE service primitive.

Table 6.1-1—A-ASSOCIATE service parameters

A-ASSOCIATE Parameter Name	Request	Indication	Response	Confirmation
Requestor Address	M	M (=)	M	M (=)
Acceptor Address	M	M (=)	M	M (=)
Data Link Service Class	M			

Blank entries are not applicable to the A-ASSOCIATE service primitives.

6.1.1.1 Requestor address

The Requestor Address identifies the Application Entity that is requesting the A-ASSOCIATE service. This address is based on the Source Addresses specified in Section 6.8.2.2. The Requestor Address specifies the DICOM Application Address and the DICOM Network Address of the requesting entity.

Note— The DICOM Application Address was called Logical Address in the previous version of this Standard.

6.1.1.2 Acceptor address

The Acceptor Address identifies the Application Entity that accepts the A-ASSOCIATE request. This address is based on the Destination Address specified in Section 6.8.2.2. The Acceptor Address specifies the DICOM Application Address and the DICOM Network Address of the accepting entity.

Note— The DICOM Application Address was called Logical Address in the previous version of this Standard.

6.1.1.3 Data link service class

The Data Link Service Class specifies the type of Data Link service to be used when transporting DICOM message fragments. Refer to section 6.8.1.1.3 of this Part of the Standard for more information concerning the use of the Data Link Service Class.

6.1.2 A-ASSOCIATE SERVICE PROCEDURE

The A-ASSOCIATE service is a confirmed service. A DICOM Application Entity (AE) uses the A-ASSOCIATE service to establish an association by issuing an A-ASSOCIATE Request primitive. The requesting and accepting AEs are identified by parameters in the Request primitive. These parameters correspond to the Source and Destination Addresses specified in Section 6.8.2.2.

The peer STN Layer provides an A-ASSOCIATE Indication primitive to the accepting AE. The accepting AE accepts or rejects the connection by sending an A-ASSOCIATE Response primitive or an A-RELEASE Request primitive.

The requesting AE must wait until it has received an A-ASSOCIATE Confirmation primitive or an A-RELEASE Indication primitive. Upon receipt of the A-ASSOCIATE Confirmation primitive, the connection is established. The requestor shall not issue any primitive except an A-RELEASE Request primitive until it receives an A-ASSOCIATE Confirmation or an A-RELEASE Indication primitive.

6.2 *Release service*

The STN Layer Release service is a confirmed service that supports the following four primitives:

a) A-RELEASE Request

This primitive is passed from the requesting Application Entity to its STN Layer in order to request that an association be terminated and the channel be closed.

b) A-RELEASE Indication

This primitive is passed from the STN Layer to the accepting Application Entity in order to inform the application that a peer wishes to terminate an association and close the channel.

c) A-RELEASE Response

This primitive is passed from the accepting Application Entity to its STN Layer in order to accept the termination of an association and closing of the channel requested by a peer.

d) A-RELEASE Confirmation

This primitive is passed from the STN Layer to the requesting Application Entity in order to inform the application that a peer has accepted the termination of the association and the closing of the channel.

6.2.1 A-RELEASE PARAMETERS

There are no required parameters for an A-RELEASE service primitive.

6.2.2 A-RELEASE SERVICE PROCEDURE

The A-RELEASE service is a confirmed service. The A-RELEASE Request primitive allows the acceptor to reject a requestor"s request for connection. It also allows either AE to break an established connection. An AE wishing to break a connection issues an A-RELEASE Request primitive to its STN Layer.

The peer STN Layer provides an A-RELEASE Indication primitive to the accepting AE. The accepting AE accepts the disconnection by sending an A-RELEASE Response primitive to its STN Layer.
The requesting AE must wait until it has received an A-RELEASE Confirmation primitive. Upon receipt of the A-RELEASE Confirmation primitive, the connection is broken. The requestor shall not issue any primitive until it receives an A-RELEASE Confirmation primitive.

Note— Even though an AE shall not issue any primitives until the A-RELEASE Confirmation is received, the STN Layer may issue multiple CCR Packets. Refer to section 6.8.1 for information about Packets.

If an A-RELEASE Request is sent in each direction, both devices shall respond with an A-RELEASE Response and the channel is closed. Both devices shall receive and ignore the A-RELEASE Confirmation on the channel after the close is effected.

6.3 *Abort service*

The A-P-ABORT service is initiated by the STN Layer and therefore supports only one primitive:

a) A-P-ABORT Indication

This primitive is used to signal the abnormal release of the association due to service problems at the STN Layer and below.

6.3-1 A-P-ABORT PARAMETERS

Table 6.3-1 lists the parameter that is required for the A-P-ABORT service.

Table 6.3-1—A-P-Abort service parameters

A-P-ABORT Parameter Name	Indication
Provider Reason	P

6.3.1.1 Provider reason

The Provider Reason parameter is used to convey the reason for aborting the association. The reason can include, but is not limited to, the following: reason not specified, unrecognized or unexpected PDU, unrecognized, unexpected or invalid PDU parameter.

Note— The generation and handling of such errors is implementation specific and therefore beyond the scope of this Standard.

6.3.2 A-P-ABORT SERVICE PROCEDURE

When the STN Layer service provider detects an internal error, an A-P-ABORT Indication primitive shall be issued. The association shall be abnormally released. The Application Entity shall issue no further primitives on the association.

6.4 *Data transfer service*

The STN Layer Data Transfer service is a non-confirmed service that supports the following two primitives:

a) P-DATA Request

This primitive is passed from the requesting Presentation Entity to its STN Layer in order to send data to its peer via the established association. The data contains Command or Data group data. The type of data group is specified in the P-DATA Request.

b) P-DATA Indication

This primitive is passed from the STN Layer to the accepting Presentation Entity in order to receive data from its peer via the established association. The data contains Command or Data group data. The type of data group is specified in the P-DATA Indication.

6.4.1 P-DATA PARAMETERS

Table 6.4-1 lists the parameters that are required for a P-DATA service primitive.

Table 6.4-1—P-DATA service parameters

P-DATA Parameter Name	Request	Indication
Command/Data Flag	M	M (=)
Last Block Flag	M	M (=)
Data Unit	M	M (=)

6.4.1.1 Command/data flag

The Command/Data flag in a P-DATA service primitive specifies whether the associated data is part of the DICOM Message Command or Data group. If this parameter is set to one (1), the data shall belong to the DICOM Command group. A value set to zero (0) indicates the data shall belong to the Data group.

6.4.1.2 Last block flag

The Last Block flag in a P-DATA service primitive indicates that the last word of a DICOM Command or Data group is in the current Data Unit. If this parameter is set to zero (0), the data unit shall not indicate the last block. A value set to one (1) shall indicate the last block.

6.4.1.3 Data unit

The Data Unit contained in the P-DATA service primitive is a Command and/or Data set of a DICOM Message.

6.4.2 P-DATA SERVICE PROCEDURE

The P-DATA service is an non-confirmed service. Either AE may cause DICOM Message Information to be exchanged. DICOM Messages shall be exchanged per the message exchange protocol as defined in Part 7 of the DICOM Standard.

Note— The message exchange rules defined in Part 7 of the DICOM Standard state that the AE requesting the connection shall also initiate the first DICOM Message exchange. The accepting AE may not initiate a DICOM Message Exchange until after the requesting AE has done so.

6.5 *Session/transport/network state machine*

6.5.1 STATE DEFINITIONS

Table 6.5-1—State definitions

State	Definition
Sta1	Idle
Sta2	Awaiting local A-ASSOCIATE Response primitive
Sta3	Awaiting peer A-ASSOCIATE Response PDU
Sta4	Association established and ready for data transfer
Sta5	Association established, ready for data transfer and paused
Sta6	Awaiting resolution of internal flow control
Sta7	Awaiting local A-RELEASE Response primitive
Sta8	Awaiting peer A-RELEASE Response PDU or S-CLOSE time-out

6.5.2 STATE ACTIONS

The state machine transition actions are broken into four categories: establishment of a connection, normal breaking of a connection, data transfer and connection aborting.

Table 6.5-2—Association establishment actions

Action	Definition
AC-1	Send A-ASSOCIATE Request PDU to peer and start S-CLOSE timer
AC-2	Issue A-ASSOCIATE Indication primitive to Application Entity
AC-3	Send A-ASSOCIATE Response PDU to peer
AC-4	Issue A-ASSOCIATE Confirmation primitive to Application Entity and stop S-CLOSE timer

Table 6.5-3—Data transfer actions

Action	Definition
DT-1	Send P-DATA Request PDU to peer
DT-2	Issue P-DATA Indication primitive to Presentation Entity
DT-3	Enter Paused state by queuing all subsequent primitives destined for peer
DT-4	Exit Paused state and send all queued primitives to peer
DT-5	Send T-PAUSE Request PDU to peer
DT-6	Send T-RESUME Request PDU to peer (If paused from peer, next state is sta5, otherwise, next state is sta4)

Table 6.5-4—Disconnection actions

Action	Definition
AR-1	Send A-RELEASE Request PDU to peer and start S-CLOSE timer
AR-2	Issue A-RELEASE Indication primitive to Application Entity
AR-3	Send A-RELEASE Response PDU to peer
AR-4	Issue A-RELEASE Confirmation primitive to Application Entity, close channel and stop S-CLOSE timer

Table 6.5-5—Abort actions

Action	Definition
AA-1	Send A-RELEASE Request PDU and start S-CLOSE timer
AA-2	Stop S-CLOSE timer, if running, and close channel
AA-3	Send A-RELEASE Request PDU, issue an A-P-ABORT Indication primitive, start S-CLOSE timer
AA-4	Ignore PDU

6.5.3 STATE TRANSITION TABLE

Table 6.5-6—Session/transport/network layer protocol state transition table

STATES	Idle	Associate		Data			Release	
EVENTS	Sta1	Sta2	Sta3	Sta4	Sta5	Sta6	Sta7	Sta8
A-ASSOCIATE Request primitive	AC-1 sta3							
A-ASSOCIATE Response primitive		AC-3 sta4						
A-ASSOCIATE Request PDU	AC-2 sta2A	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8
A-ASSOCIATE Response PDU	AA-4 sta1	AA-3 sta8	AC-4 sta4	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8
P-DATA Request primitive				DT-1 sta4	DT-3 sta5	DT-1 sta6		
P-DATA Request PDU	AA-4 sta1	AA-3 sta8	AA-3 sta8	DT-2 sta4	DT-2 sta5	DT-2 sta6	AA-3 sta8	AA-4 sta8

Table continued on next page

Table 6.5-6 continued

A-RELEASE Request primitive		AR-1 sta8	AR-1 sta8	AR-1 sta8	AR-1 sta8	AR-1 sta8	AA-3 sta8	
A-RELEASE Response primitive							AR-3 sta8	
A-RELEASE Request PDU	AA-4 sta1	AR-2 sta7	AR-2 sta7	AR-2 sta7	AR-2 sta7	AR-2 sta7	AA-4 sta7	AR-3 sta8
A-RELEASE Response PDU	AA-4 sta1	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8	AR-4 sta1
Internal Flow Control PAUSE				DT-5 sta6	DT-5 sta6			
Internal Flow Control RESUME						DT-6 sta4 sta5		
T-PAUSE PDU	AA-4 sta1	AA-3 sta8	AA-3 sta8	DT-3 sta5	AA-4 sta5	DT-3 sta6	AA-1 sta8	AA-1 sta8
T-RESUME PDU	AA-4 sta1	AA-3 sta8	AA-3 sta8	AA-4 sta4	DT-4 sta4	DT-4 sta6	AA-1 sta8	AA-1 sta8
S-CLOSE timer expired			AA-2 sta1					AA-2 sta1
Unknown PDU	AA-4 sta1	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8	AA-3 sta8

6.6 *Exception handling*

The STN Layer user aborts a non-reconstructible or uninterpretable message fragment by sending an A-RELEASE Request primitive. Refer to Part 7 of the DICOM standard for information about actions taken when an exception occurs during transmission or reception of a DICOM message.

Note— As Part 7 of the DICOM standard states, if a connection is closed before the complete message was sent, the sender shall assume that no portion of the message was received. If a connection is closed before the complete message was received, the receiver shall act as if no portion of the message was received.

Unresolved errors in the STN Layer are handled by releasing the association as described in section 6.2.2 of this Part and then closing the channel.

6.7 *Flow control*

A receiving STN Layer handles flow control by sending a T-PAUSE Request PDU to its peer in order to stop the peer from sending PDUs. A T-RESUME Request PDU is used by the receiver STN Layer to request that the sender resume sending PDUs. T-PAUSE and T-RESUME Request PDUs are valid only when in the data transfer state. A sender that gets tired of waiting may prematurely terminate the message transmission by releasing the association as described in 6.5. The A-RELEASE Request PDU is the only PDU that may be transmitted by the sender after it receives a T-PAUSE Request PDU and before it receives a T-RESUME Request PDU.

When an A-ASSOCIATE or an A-RELEASE Request primitive is issued, the receipt of the appropriate confirmation is expected. No subsequent request should be made of the peer until the expected confirmation is received. (The one exception to this rule is an A-RELEASE Request when waiting for confirmation of the A-ASSOCIATE Request.)

There is no specified time-out interval for the receipt of a confirmation. A peer may withhold sending a confirmation for as long as it needs. This will essentially keep the association requestor from sending messages until the peer is capable of receiving them. If the issuer of an A-ASSOCIATE Request primitive decides that it has waited long enough for its confirmation, it shall issue an A-RELEASE Request primitive to inform the peer that it no longer desires an association.

6.8 *Session/transport/network data unit structure*

In this Part of the Standard, the session, transport and network layers have been presented as one protocol layer. The functionality of each of these layers, however, must still be performed. This section identifies the required session, transport and network protocols needed to transmit and receive DICOM messages.

The STN Layer defines a header that is required on all data units that are sent to or received from the Data Link Layer. This STN header is comprised of three fields: the Packet (transport/network layer) header, the Connection (session layer) header and data. Figure 6.8-1 describes the STN header. Every STN protocol (PDU) consists of an STN header. The required fields of an STN header differ with the type of PDU being sent. Sections 6.8.3 thru 6.8.9 defined the format of each of the STN PDUs.

Packet Header
Connection Header
Data

Figure 6.8-1—STN PDU header

6.8.1 PACKET HEADER

The information required to support the transport and network layers is defined in the fields of the Packet Header. The Packet header is made up of two parts: the Packet Descriptor Word and the Block Sequence Word. Figure 6.8-2 illustrates the Packet Header structure. Each STN PDU requires a Packet Header. The Packet Header is always transmitted in Little Endian byte order.

15	14		0
Packet Descriptor Word			
Lastblk	Block Sequence Number		

Figure 6.8-2—Packet header format

6.8.1.1 Packet descriptor word

The Packet Descriptor Word contains a Packet Type field, a Channel Number field and a Service Class field. The format of this word is illustrated in Figure 6.8-3. The component fields are described in the following sections.

15 14 13 12 11 10 9 8	7 6 5 4	3 2 1 0
Channel Number	Type	Service Class

Figure 6.8-3—Packet descriptor word

6.8.1.1.1 Packet type field

The Packet Type field of the Packet Header is used to specify the type of packet being delivered. The various packet types include Data, Command or Transport Control. Eight types of packets are used to provide peer-to-peer communication. Table 6.8-2 defines the Packet Type bit encoding.

Table 6.8-2—Packet type codes and definitions

Code	Name and Description
1010	Close Channel Request (CCR). This code requests release of a channel with abandonment of any pending data or refusal to accept an Open Channel Request or other Packet Type.
1000	Close Channel Indication (CCI). This is an acknowledgement of a Close Channel Request.
0110	Open Channel Request (OCR). This code requests that a virtual channel be established. The Service Class field specifies the requested service class. Session layer information containing installation-specific addresses are included in the Data field.
0100	Open Channel Indication (OCI). This is the acknowledgement to an Open Channel Request. The Connection Header information shall concur.
0010	Command (CMD). These packets types contain message command group information. See Note below.
0000	Data (DATA). These packets contain from 0 to 2048 words of message data.
1001	Pause Request (PAUSE). This request can be sent by an STN entity to halt data transmission on the specified channel until a RESUME is received for that channel.
0101	Resume Request (RESUME). This request is sent by an STN entity that has previously sent a PAUSE to indicate that transmission of message packets may resume on the specified channel.

NOTES— 1. Command group information includes only those elements in groups 0 and 1. Refer to Part 7 of the DICOM Standard for more information concerning Command and Data groups.

2. The usage of the word "indication" in section 6.8.1.1.1 does not conform strictly to the ISO usage. An "indication" described above is really "response" in the ISO usage. The term "indication" is used here to maintain conformance with the ACR-NEMA 300-1988 Standard.

6.8.1.1.2 Channel number field

The Channel Number field contains an 8-bit channel number. The Channel Number used when establishing a connection is implementation specific. The only limits placed on its value are:

— when attempting to establish a new association, an unused (closed) channel must be selected

— support of channel 0 is required

NOTE— Each simultaneous association established shall use a different Channel Number. There is a one-to-one relationship between an association and a Channel Number.

6.8.1.1.3 Service class field

The Service Class field is provided to permit the association (channel) requestor to select the type of Data Link Layer service to be used when transporting DICOM Message fragments. Table 6.8-3 lists and specifies the Service Class bit encoding. The specified Data Link Layer service class is only relevant for DATA and CMD packet types. The Data Link Layer service used for communicating Control packets for a given channel may be different from that specified for the DATA and CMD packet types.

Table 6.8-3—Service class codes and definitions

Code	Service Class
xxx1	DATA - No data link acknowledgements. Data packets will be sent in DATA frames
xxx0	DATA/ACK - Data link acknownledgements required. Data packets will be sent in DATA/ACK frames. (Default)

Bits 1-3 of the Service Class Code are reserved for future use. These bits shall be set to zero by the sender and not checked by the receiver. The default service class is DATA/ACK and shall be supported. DATA service support is optional.

If a peer A-ASSOCIATE Request PDU is to be refused by the acceptor with an A-RELEASE Request PDU, then the A-RELEASE Request PDU (Close Channel Request) must specify a service class supported by the refusing device.

6.8.1.2 Block sequence word

The Block Sequence Word is composed of two fields: the Block Sequence Number and the Last Block Flag. Figure 6.8-2 illustrates this word.

The Block Sequence Number (BSN) Field is used to sequence a series of DATA or CMD packets in a DICOM message. The BSN is a 15-bit value and is incremented by one for each succeeding packet of the message segment. This field is zero for all other packet types.

The most significant bit of the Block Sequence Word is reserved as the Last Block Flag bit. This flag shall be asserted (set) only for the last block of a Command or Data message segment.

When receiving P-DATA Request PDUs, the BSN must be equal to zero if it is the first PDU received, or one greater than the preceding PDU"s Block Sequence Number. If the BSN is not as such, the protocol shall be in error and the channel shall be closed and the association ended as described in 6.5 of this Part.

The first block of a Command segment shall be numbered zero. The last block of a Command segment shall be numbered properly and the Last Block Flag shall be set. If a Data segment follows the Command segment in a message, the first block of the Data segment shall be numbered zero. The last block of a Data segment shall be numbered properly and the Last Block Flag shall be set.

6.8.2 CONNECTION HEADER

The information needed to support a session layer protocol is defined in the fields of the Connection Header. Figure 6.8-2 illustrates the Connection Header Format. This header is comprised of a Connection Control Word, a message Destination Address and a message Source Address. Each STN PDU includes some subset of the Connection Header. (This subset may be the NULL subset, ie, none of the Connection Header fields are present.)

The full Connection Header is 33 16-bit words in length. The Connection Header, or any subset, is transferred in Little Endian format. The Application Addresses shall be specified as text strings and the Network Addresses shall be specified as binary integers.

6.8.2.1 Connection control word

The Connection Control Word is a single 16-bit word and the first word of the Connection Header. The upper 12 bits of this word are reserved and each is set to zero. Table 6.8-3 defines the usage of the lower four bits of this word.

6.8.2.2 Destination/source fields

Connect Request and Connect Response control words shall contain source and destination data in the format specified in Figure 6.8-4. The format of the source and destination fields shall be as follows: Destination Application Address, Destination Network Address, Source Application Address, and Source Network Address. Each set of addresses is eight words long.

NOTE— The DICOM Application Address was called Logical Address in the previous version of this Standard.

The destination fields will always refer to the receiver of the PDU, and the source fields to the sender.

NOTE— This means that the source fields of an A-ASSOCIATE Request PDU will be used as the destination fields of the confirming A-ASSOCIATE Response PDU, and the source fields of the A-ASSOCIATE Response PDU will be those of the sender.

It is recommended that the application addresses used in these fields be the same as those used at the application layer. Each of the four addresses is required to contain pertinent information. There are no default Application or Network Addresses.

15 8	7 0
Connection Control Word	
Destination Application Addr Byte 2	Destination Application Addr Byte 1
Destination Application Addr Byte 4	Destination Application Addr Byte 3
Destination Application Addr Byte 6	Destination Application Addr Byte 5
Destination Application Addr Byte 8	Destination Application Addr Byte 7
Destination Application Addr Byte 10	Destination Application Addr Byte 9
Destination Application Addr Byte 12	Destination Application Addr Byte 11
Destination Application Addr Byte 14	Destination Application Addr Byte 13
Destination Application Addr Byte 16	Destination Application Addr Byte 15
Destination Network Address Word 1 Destination Network Address Word 2 Destination Network Address Word 3 Destination Network Address Word 4 Destination Network Address Word 5 Destination Network Address Word 6 Destination Network Address Word 7 Destination Network Address Word 8	
Source Application Address Byte 2	Source Application Address Byte 1
Source Application Address Byte 4	Source Application Address Byte 3
Source Application Address Byte 6	Source Application Address Byte 5
Source Application Address Byte 8	Source Application Address Byte 7
Source Application Address Byte 10	Source Application Address Byte 9
Source Application Address Byte 12	Source Application Address Byte 11
Source Application Address Byte 14	Source Application Address Byte 13
Source Application Address Byte 16	Source Application Address Byte 15
Source Network Address Word 1 Source Network Address Word 2 Source Network Address Word 3 Source Network Address Word 4 Source Network Address Word 5 Source Network Address Word 6 Source Network Address Word 7 Source Network Address Word 8	

Figure 6.8-4—Connection header format

Table 6.8-3—Connection control word definition

Reserved Field	Defined Bits	Name and Description
0000 0000 0000	0110	Connect Request. This control word is used to request an end-to-end connection for sending a message to a specified destination. It is contained in an A-ASSOCIATE Request PDU along with the Message Source and Destination Addresses as defined in 6.8.2.2.
0000 0000 0000	0100	Connect Response. This control word is used to acknowledge an end-to-end Connect Request. It is contained in an A-ASSOCIATE Response PDU along with the Message Source and Destination Addresses as defined in 6.8.2.2.
0000 0000 0000	1010	Disconnect Request. This control word is used to break a connection. It is contained in an A-RELEASE Request PDU. The Message Source and Destination Addresses are not transmitted with the Disconnect Request.
0000 0000 0000	1000	Disconnect Response. This control word is used to acknowledge a Disconnect Request. It is contained in an A-RELEASE Response PDU. The Message Source and Destination Addresses are not transmitted with the Disconnect Response.

6.8.3 A-ASSOCIATE REQUEST PDU

The A-ASSOCIATE Request PDU shall be made of a sequence of mandatory fields. Table 6.8-4 shows the sequence of the mandatory fields.

Table 6.8-4—A-ASSOCIATE request PDU fields

Item words	Field Name	Description
1	Packet Descriptor Word	The Packet Descriptor word contains the number of the channel to be opened in the Channel Number Field, the Open Channel Request bit encoding in the Packet Type Field and the service class specification. Refer to section 6.8.1.1.
2	Block Sequence Number	The Block Sequence Number shall be zero and the Last Block Flag shall be set as specified in section 6.8.1.2 for this PDU.
3	Connection Control Word	0006H
4-11	Destination Application Address	The Destination Application Address specifies the name of the accepting application and shall be encoded as described in section 6.8.2.2.
12-19	Destination Network Address	The Destination Network Address specifies the name of the acceptor and shall be encoded as described in section 6.8.2.2.
20-27	Source Application Address	The Source Application Address specifies the name of the requesting application and shall be encoded as described in section 6.8.2.2.
28-35	Source Network Address	The Source Network Address specifies the name of the requestor and shall be encoded as described in section 6.8.2.2.

6.8.4 A-ASSOCIATE RESPONSE PDU

The A-ASSOCIATE Response PDU shall be made of a sequence of mandatory fields. Table 6.8-5 shows the sequence of the mandatory fields.

Table 6.8-5—A-ASSOCIATE response PDU fields

Item words	Field Name	Description
1	Packet Descriptor Word	The Packet Descriptor word contains the number of the channel that has been opened in the Channel Number Field, the Open Channel Indication bit encoding in the Packet Type Field and the service class specification specified by the requestor. Refer to section 6.8.1.1.
2	Block Sequence Number	The Block Sequence Number shall be zero and the Last Block Flag shall be set as specified in section 6.8.1.2 for this PDU.
3	Connection Control Word	0004H
4-11	Destination Application Address	The Destination Application Address specifies the name of the application requesting the association and shall be encoded as described in section 6.8.2.2.
12-19	Destination Network Address	The Destination Network Address specifies the name of the association requestor and shall be encoded as described in section 6.8.2.2.
20-27	Source Application Address	The Source Application Address specifies the name of the application accepting the association request and shall be encoded as described in section 6.8.2.2.
28-35	Source Network Address	The Source Network Address specifies the name of the acceptor and shall be encoded as described in section 6.8.2.2.

6.8.5 A-RELEASE REQUEST PDU

The A-RELEASE Request PDU shall be made of a sequence of mandatory fields. Table 6.8-6 shows the sequence of the mandatory fields.

Table 6.8-6—A-RELEASE request PDU fields

Item words	Field Name	Description
1	Packet Descriptor Word	The Packet Descriptor word contains the number of the channel to be closed in the Channel Number Field, the Close Channel Request bit encoding in the Packet Type Field and the service class specification associated with this channel. Refer to section 6.8.1.1.
2	Block Sequence Number	The Block Sequence Number shall be zero and the Last Block Flag shall be set as specified in section 6.8.1.2 for this PDU.
3	Connection Control Word	000AH

6.8.6 A-RELEASE RESPONSE PDU

The A-RELEASE Response PDU shall be made of a sequence of mandatory fields. Table 6.8-7 shows the sequence of the mandatory fields.

Table 6.8-7—A-RELEASE response PDU fields

Item words	Field Name	Description
1	Packet Descriptor Word	The Packet Descriptor word contains the number of the closed channel in the Channel Number Field, the Close Channel Indication bit encoding in the Packet Type Field and the service class specification associated with this channel. Refer to section 6.8.1.1.
2	Block Sequence Number	The Block Sequence Number shall be zero and the Last Block Flag shall be set as specified in section 6.8.1.2 for this PDU.
3	Connection Control Word	0008H

6.8.7 P-DATA REQUEST PDU

The P-DATA Request PDU is used to transmit CMD and DATA packet types. There are two kinds of P-DATA Request PDUs. One PDU transmits CMD packets and the other transmits DATA packets. The P-DATA PDU required is specified by the information received in the P-DATA Request primitive from the Application Entity. Each PDU is made of a sequence of mandatory fields plus one variable length field. The two P-DATA Request PDUs are described in the following section.

6.8.7.1 P-DATA command request PDU

The P-DATA Command Request PDU shall be made of a sequence of mandatory fields and one variable length field. Table 6.8-8 shows the sequence of these fields.

Table 6.8-8—P-DATA command request PDU fields

Item words	Field Name	Description
1	Packet Descriptor Word	The Packet Descriptor word contains the number of the channel with which the data is associated in the Channel Number Field, the Command Request bit encoding in the Packet Type Field and the service class specification associated with this channel. Refer to section 6.8.1.1.
2	Block Sequence Number	The Block Sequence Number is set to the packet sequence number and the Last Block Flag is set if this PDU is the last one in the Command message and cleared otherwise. Refer to section 6.8.1.2.
3-xxx	Data	0 to 2048 words of Command data.

6.8.7.2 P-DATA data request PDU

The P-DATA Data Request PDU shall be made of a sequence of mandatory fields and one variable length field. Table 6.8-9 shows the sequence of these fields.

Table 6.8-9—P-DATA data request PDU fields

Item words	Field Name	Description
1	Packet Descriptor Word	The Packet Descriptor word contains the number of the channel with which the data is associated in the Channel Number Field, the Data Request bit encoding in the Packet Type Field and the service class specification associated with this channel. Refer to section 6.8.1.1.
2	Block Sequence Number	The Block Sequence Number is set to the packet sequence number and the Last Block Flag is set if this PDU is the last one in the Data message and cleared otherwise. Refer to section 6.8.1.2.
3-xxx	Data	0 to 2048 words of Data data.

6.8.8 T-PAUSE REQUEST PDU

The T-PAUSE Request PDU is sent by to a peer STN Layer requesting that the transmission of PDUs for a specified channel be suspended. The receiving STN entity is expected to refrain from sending any further PDUs until it receives a T-RESUME Request PDU from its peer specifying the same channel number. The T-PAUSE Request PDU shall be made up of a series of mandatory fields. These fields are described in Table 6.8-10.

Table 6.8-10—T-PAUSE request PDU fields

Item words	**Field Names**	**Description**
1	Packet Description Word	The Packet Descriptor word contains the number of thte channel to be paused in the Channel Number Field, the Pause bit encoding in the Packet Type Field and the service class specification associated with the channel. Refer to section 6.8.1.1.
2	Block Sequence Number	The Block Sequence Number shall be zero and the Last Block Flag shall be set as specified in section 6.8.1.2 for this P.D.U.

6.8.9 T-RESUME REQUEST PDU

The T-RESUME Request PDU is sent to a peer STN entity to request that the transmission of PDUs for a specified channel be resumed. It is assumed that a T-PAUSE Request PDU was previously sent. The peer STN Layer, upon reception of this PDU, shall begin sending PDUs for the specified channel. The T-RESUME Request PDU shall be made up of a series of mandatory fields. These fields are described in Table 6.8-11.

Table 6.8-11—T-RESUME request PDU fields

Item words	Field Name	Description
1	Packet Descriptor Word	The Packet Descriptor word contains the number of the channel to be resumed in the Channel Number Field, the Resume bit encoding in the Packet Type Field and the service class specification associated with this channel. Refer to section 6.8.1.1.
2	Block Sequence Number	The Block Sequence Number shall be zero and the Last Block Flag shall be set as specified in section 6.8.1.2 for this PDU.

7 Point to point data link layer protocol

The information contained in this section provides a description of the Data Link Layer service and protocol for the DICOM Point to Point message exchange. The Data Link Layer provides for the transmission of frames across the interface. The Data Link Layer service is used by the STN Layer to effect transport of STN Layer PDU"s.

All data sent across the interface is enclosed in a frame. A frame contains a descriptor word, an STN Layer PDU (packet) and a checksum. The Data Link Layer must allow for interface arbitration after transmission of each frame. Flow control and exception handling of frames is detailed in Sections 7.2 and 7.3, respectively.

It is possible to conform to the DICOM point to point message exchange while only supplying or using a subset of the services defined in this section. Section 9 details the conformance requirements.

NOTE— The terms PDU and frame are used interchangeably in this section.

There is one Data Link Layer service provided by the Data Link Layer: data transfer. Table 7-1 lists this service.

Table 7-1—Data link layer services

SERVICE	TYPE
STN-DATAN	on-confirmed

7.1 *Stn-data service*

The STN-DATA service allows for the transport of STN Layer control and data packets (PDUs). The parameters given in an STN-DATA request shall indicate the type of service class provided over the channel. The service class can specify DATA or DATA/ACK type.

NOTES— 1. DATA service class should be used when high data transport rates are required. This service is more susceptible to flow control and exception handling problems then the DATA/ACK service class.
2. Supporting the DATA service class is optional.

The STN-DATA service is a non-confirmed service and therefore contains only two primitives:

a) STN-DATA Request

This primitive is passed from the STN Layer to its Data Link Layer in order to send data to the peer. There is no assumption about the contents of the data.

b) STN-DATA Indication

This primitive is passed from the Data Link Layer to its STN Layer when receiving data from a peer. There is no assumption about the contents of the data.

7.1.1 STN-DATA PARAMETERS

Table 7.1-1 lists the STN-DATA service parameters.

Table 7.1-1—STN-DATA service parameters

STN-DATA Parameter Name	Request	Indication
Data Service Type	M	
Data Unit	M	M(=)

7.1.1.1 Data service type

The Data Service type in a STN-DATA Request primitive specifies the type of service being performed over the channel. The service type can specify DATA or DATA/ACK type as defined in Section 6.8.1.1.3. This service type shall be employed by the Data Link Layer when transmitting the frame to its peer.

7.1.1.2 Data unit

The Data Unit contained in the STN-DATA Request primitive is an STN Layer packet that is to be sent to the peer.

7.1.2 STN-DATA SERVICE PROCEDURE

The requesting STN Layer may issue an STN-DATA Request primitive at anytime to request the transfer of an STN packet. The requesting Data Link Layer shall place the specified Data Unit in a data frame (either DATA or DATA/ACK depending upon the service class specified) and transmit it to the peer Data Link Layer.

If the service class is DATA/ACK, the accepting Data Link Layer shall return a D-ACK or D-NACK PDU to acknowledge reception of the frame. The accepting Data Link Layer shall issue an STN-DATA indication primitive to its STN Layer upon error free reception of the data frame.

If the service class is DATA, the accepting Data Link Layer shall issue an STN-DATA indication primitive to its STN Layer upon error free reception of the data frame.

NOTE— The lack of error recovery in the DATA service makes it a poor choice for STN Layer Control Packet transmission. Note also that since the transmitter may be faster than the receiver, DATA service flow control is not assured. DATA service should only be used when a receiver is guaranteed by system architecture to be faster than the transmitter.

If the Data Link Layer detects a Physical Layer Bit Error in a received STN-DATA service frame, it shall not be required to report the error to the peer Data Link Layer. The received frame shall be ignored. The STN Layer will detect the missing frame.

A Data Link Layer may attempt to flow control the reception of data frames. Flow control may begin by either asserting the Physical Layer INTO signal or by transmitting a Data Link Layer D-STATUS Request PDU. A Data Link Layer D-STATUS Request PDU would be sent by the flow control requestor, reporting a status indicating no buffers available.

7.2 *Exception handling*

The Data Link Layer must handle data errors as determined by the parity and frame check sequence fields. The D-NACK frame reports these errors and elicits a retransmission. A D-STATUS Request frame can be sent by the acceptor if it is not expected to acknowledge the frame or by the requestor if its retry count has been exceeded. The Data Link Layer enforces the specified frame sequence protocol (Figure 7.5-1).

If a frame is received that is erroneous or cannot be interpreted, the acceptor will send a D-STATUS Request or a D-NACK frame indicating the error, depending on the received frame type. If a D-ACK frame was expected, the returned D-STATUS Response frame will contain the error status of the previous frame.

If an interrupt occurs (via INTO or INTI) during the transmission of a frame, the receiver will discard the frame fragment and the sender is expected to resend the total frame. The retransmission will occur after the resolution of the interrupt condition. To resolve the interrupt condition, the receiver of the interrupt shall initiate a D-STATUS Request PDU and wait for the D-STATUS Response PDU. The following conditions might result in the initiation of an interrupt.

a) A time specified for sending data (Figure 8.2-4) exceeded the maximum value of 1 second. (Physical Layer Time-out)

b) The physical state sequence protocol is violated (Figure 8.3-1). (Unspecified Error at Data Link)

c) Flow control and exception handling for DATA service class.

d) A hardware reset has occurred.

Refer to Appendix C for more information about exception handling.

7.3 *Flow control*

The Data Link Layer can exert flow control on data sent in D-DATA_ACK frames. When a D-DATA_ACK frame is sent, the reception of a D-ACK or D-NACK frame is always the expected response. There should be no time-out interval for the reception of these response frames. Therefore, sending another frame may be controlled by the receiver by delaying the response frame. The reception of a D-ACK or D-NACK frame is the signal that the acceptor is prepared to receive another frame.

If a requestor waiting for a D-ACK frame has reached an internal-specific time-out and needs to determine the transmission status, it can inquire by sending a D-STATUS Request frame with Acknowledge Pending Waiting for D-ACK status.

Both devices must be able to receive the two-word D-STATUS Request, D-STATUS Response, D-ACK and D-NACK frames at all times. D-DATA, D-DATA_ACK and D-ECHO Request and Response frames can also arrive at any time (except as noted above for D-DATA_ACK.) These shall only be refused by interrupting the Physical Layer.

Flow control for both D-DATA_ACK and D-DATA frames can be accomplished as follows: The receiver can interrupt the sender and the sender can then request status. The receiver will then respond with status indicating a buffer unavailable condition. To resume transmission, the receiver should send a D-STATUS Request indicating normal conditions. If the interrupt occurs during transmission of a D-DATA frame, that frame should be retransmitted once the data flow has been resumed as specified in 7.2.

7.4 *Interface off-line*

Upon the occurrence of an interface failure, such as system reset or a power failure, the interface must recover in a predetermined fashion. Since there can be no prediction as to the state of the interface, the recovery must begin at the Physical Layer. Therefore, INTO is to be asserted by the interface immediately after initialization. This will cause the peer to initiate a D-STATUS Request PDU. The resetting interface will respond with the Interface Not Serviced status in the D-STATUS Response PDU. This status causes the peer to close down its interface and to inform its upper layers that all channels have been closed. After the resetting interface is functional, it will issue a D-STATUS Request PDU to the peer indicating No Error, Interface Available. At this point, normal communication may begin.
Figure 7.4-1 shows this sequence of events.

Interface-A		**Interface-B**
	---->	INTI
INTO	<----	D-STATUS Request (No Error)
D-STATUS Response (Interface Not Serviced)	---->	
D-STATUS Request (No Error)	---->	
		D-STATUS Response (No Error)

Figure 7.4-1 Interface off-line recovery

7.5 *Data link state machine*

The state diagram for the Data Link Layer details the state transitions permitted in the Data Link Layer. The Data Link state diagram is shown in Figure 7.5-1.

NOTE— The term "indicate" in the Data Link state diagram really denotes a "response". The terminology has been maintained to be consistent with previous versions of this Standard.

Frame Sequence Under Normal Conditions

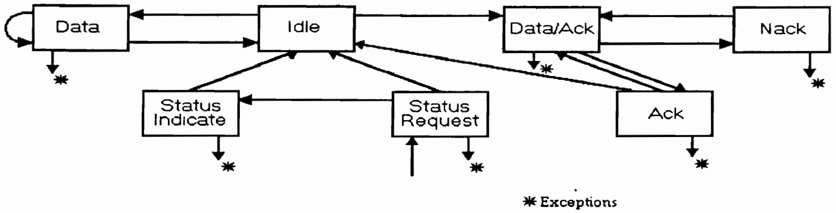


Diagram of Allowed State Sequences

Previous Frame \ Next Frame	Data	Data/Ack	Ack	Nack	Status Request	Status Indicate	Idle	Echo Request	Echo Indicate
Data	Data	Data/Ack			Status Request		Idle	Echo Request	
Data/Ack			Ack	Nack	Status Request				
Ack	Data	Data/Ack			Status Request		Idle	Echo Request	
Nack	Data	Data/Ack			Status Request		Idle	Echo Request	
Status Request					Status Request	Status Indicate	Idle		
Status Indicate	Data	Data/Ack	Ack	Nack	Status Request		Idle	Echo Request	
Idle	Data	Data/Ack			Status Request		Idle	Echo Request	
Echo Request					Status Request				Echo Indicate
Echo Indicate	Data	Data/Ack			Status Request		Idle	Echo Request	

Figure 7.5-1 Data link protocol state diagrams

7.6 *Data link data unit structures*

The Data Link Layer supports eight Protocol Data Units (PDUs). These PDUs are defined in terms of Data Link Layer frames. A frame consists of a Frame Descriptor Word followed by an optional Session/Transport/Network Layer Packet and then by a Frame Check Sequence. See Figure 7.6-1
The Frame Descriptor Word and Frame Check Sequence are each 16-bit words. Each of these shall be transferred in Little Endian byte order to the peer.

15 … 0
Frame Descriptor Word
STN Layer Packet
Frame Check Sequence

Figure 7.6-1 Frame structure

7.6.1 FRAME DESCRIPTOR WORD

The Frame Descriptor Word is the first word of a Data Link PDU and it describes the contents of the frame. The Frame Descriptor Word contains a frame type field and a parameter field, as show in Figure 7.6-2. The frame type names and parameters are summarized in Table 7.6-1.

15	12	11	0
Frame Type	Parameter		

Figure 7.6.1-1 Frame descriptor word

Table 7.6-1—Frame descriptor word summary

Name	Parameter Field Contents
D-DATA Request PDU	Frame word count
D-DATA_ACK Request PDU	Frame word count
D-ACK PDU	Status
D-NACK PDU	Status
D-STATUS Request PDU	Status
D-STATUS Response PDU	Status
D-ECHO Request PDU	Frame word count
D-ECHO Response PDU	Frame word count
Reserved	-

7.6.1.1 Frame type field

The frame types and codes are defined in Table 7.6-3. All implementations of the Data Link Layer protocol must support all defined frame types, with the exception of the optional D-DATA Request PDU. Data Link frames are used to communicate information across the interface. Figure 7.5-1 contains a Data Link protocol state diagram showing all allowed frame sequences. The D-STATUS Request and Response frame types are used to communicate interface conditions or resolve problems relevant to Data Link activity. Specifically, if a data transfer is interrupted by the hardware interrupt circuits (see Section 8.2.2), then the interrupted device shall send a D-STATUS Request frame including its status, and the interrupting device must respond with a D-STATUS Response frame including the status conditions pertinent to the interrupt. If the interrupting device does not respond to the D-STATUS Request frame in an implementation specific time-out (not less than one second), it is assumed to be non-operational.

If a D-ECHO Request frame arrives, the Data Link Layer will replace the frame type with the D-ECHO Response type , recompute the frame check sequence and send the frame back to its peer.

Table 7.6-3—Frame types and codes

Name	Code	Description
D-DATA Request PDU	0000	A frame that contains an STN packet. These frames shall not be acknowledged nor not-acknowledged. The Parameter field shall contain the total frame word count.
D-DATA_ACK Request PDU	0100	A frame that contains an STN packet. The receiving Data Link must respond with a D-ACK PDU or a D-NACK PDU. The Parameter field shall contain the frame word count.
D-ACK PDU	1000	A two-word frame that shall be sent to indicate that the previous frame was received with no parity errors and the frame check sequence computed by the receiving Data Link shall be the same as that contained in the received frame. It shall contain the current status as defined in Table 7.6-4.
D-NACK PDU	1001	A two-word frame that shall be sent to indicate that the previous frame was not received successfully. The previous frame shall be retransmitted unless the sender's retry count has been exceeded, in which case a D-STATUS Request frame shall be sent. It shall contain the current status as defined in Table 7.6-4.
D-STATUS Request PDU	1101	A two-word frame sent to request a D-STATUS Response PDU. It shall contain the current status as defined in Table 7.6.1.2-1. The receiving Data Link shall respond with a D-STATUS Response PDU.
D-STATUS Response PDU	1100	A two-word frame sent in response to a D-STATUS Request PDU. It shall contain the current status as defined in Table 7.6-4.
D-ECHO Request PDU	0011	A frame that may contain a packet. The Parameter field shall contain the frame word count. The receiving Data Link shall not send a D-ACK nor a D-NACK, but shall return the packet in a D-ECHO Response PDU.
D-ECHO Response PDU	0010	A frame that contains a packet that was received in a D-ECHO Request PDU. The Parameter field shall contain the frame word count.

7.6.1.2 Frame parameter field

The parameter field of the Frame Descriptor Word holds addition information concerning the frame. This field may contain a word count or status. The word count is the total number of words that are in the frame (see Figure 1-1). The status is a bit field that specifies a Data Link Layer"s current errors and/or state. Table 7.6-4 defines the status bits and their meaning. The sections that follow further describe each bit.

Table 7.6-4—Frame status values

Frame Status (bits 0-11)	Status Description11 00
0 x x x x x x x x x x 1	Parity error
0 0 x x x x x x x x 1 x	Frame check sequence error
0 0 x x x x x x x 1 x x	Previous frame unrecognized
0 0 x x x x x x 1 x x x	Frame incomplete
0 0 x x x x x 1 x x x x	Physical Layer protocol time-out
0 0 x x x x 1 x x x x x	Data Link Layer retry count exceeded
0 0 x x x 1 x x x x x x	Acknowledge pending; waiting for D-ACK
0 0 x x 1 x x x x x x x	Unspecified error at Data Link layer
0 0 x 1 x x x x x x x x	Receive buffer unavailable
0 0 1 x x x x x x x x x	Interface not serviced
0 0 0 0 0 0 0 0 0 0 0 0	No errors, interface available

7.6.1.2.1 Parity error

The last frame received contained at least one occurrence of a parity error. This status bit should be set in a D-NACK PDU when using DATA/ACK service class or in a D-STATUS Request or Response PDU.

7.6.1.2.2 Frame check sequence error

The last frame received contained an incorrect Frame Check Sequence code for the data contained in the frame. This status should be sent in a D-NACK PDU when using DATA/ACK service class or in a D-STATUS Request or Response PDU.

7.6.1.2.3 Unrecognized frame

The last frame received did not consist of a recognizable type. Since the service class of the frame should also be in question, this error should only be reported with a D-STATUS Request or Response. In this case, the D-STATUS Request PDU effectively substitutes for the D-NACK PDU that might normally be expected when using DATA/ACK service class. After responding with a D-STATUS Response PDU, the sender should attempt to resend the original frame. This status should be used any time a frame is unrecognized, including a D-STATUS Request PDU.

7.6.1.2.4 Incomplete frame

The last frame received contained less than the number of words found in the Frame Word Count field. This status should be sent in a D-NACK PDU when using DATA/ACK service class or in a D-STATUS Request or Response PDU.

7.6.1.2.5 Physical layer time-out

A physical layer protocol time-out occurred. This error indicates that there was a period of inactivity at the physical layer in excess of 1 second during a frame transfer. This error should only be set in D-STATUS Request and Response PDUs.

7.6.1.2.6 Retry count exceeded

The Data Link retry count for resending the previous frame was exceeded. The number of retries is implementation specific, but a minimum of 3 is recommended. This status should only be set by the sender of a frame. The Data Link Layer should also pass this information to the upper layers for resolution. This error should only be used in D-STATUS Request and Response PDUs.

7.6.1.2.7 Acknowledge pending

This status should be issued by the sender of a DATA/ACK frame after the sender"s acknowledgement timer has expired. This status should only be associated with a D-STATUS Request PDU. In the event that this status is received by the receiver before the D-ACK or D-NACK PDU is sent, the receiver should respond with a Receive Buffer Unavailable status. If any other status is returned, it is assumed that the receiver has somehow "lost" the frame and it must be retransmitted. If the Acknowledge Pending status is received and no DATA/ACK frame preceded the D-STATUS Request, it can be assumed by the receiver that the sender is somehow in error. In this case, the receiver should respond with an Unspecified Error to allow the sender to recover.

7.6.1.2.8 Unspecified error

This status is presented when a Data Link protocol or Physical Layer sequence error is detected. It should force its receiver and sender to return to the idle state (see Figure 7.5-1) and resend the last packet. This error type should only be used with D-STATUS Request and Response PDUs.

7.6.1.2.9 Unavailable receive buffer

This status is returned by a receiver if it is unable to receive any additional packets, with the exception of a D-STATUS Request, which must always be accepted. There are four ways for it to be used:

a) This status may be returned in a D-ACK PDU when using DATA/ACK service class to indicate that no new buffer is available but the last packet was accepted.

b) This status may be returned in a D-NACK PDU when using DATA/ACK service class to indicate that no new buffer is available and the last packet should be resent.

c) This status may appear in a D-STATUS Response PDU as a response to a D-STATUS Request PDU with the Acknowledge Pending status set to indicate that the D-ACK/D-NACK is forthcoming.

d) This status may appear in a D-STATUS Request or Response PDU during a DATA service class transfer to allow the receiver to tell the sender that it has no buffer. This may typically be initiated by an interrupt. In the case of an interrupt, the last frame should be resent.

7.6.1.2.10 Interface not serviced

This status indicates that the upper layers of this node are inactive. The Interface Not Serviced bit should be set to indicate that an interface is down and not able to communicate. If this status is received, the receiver must inform the upper layers so that all channels can be closed. When the upper layers become active, this status should be cleared and a D-STATUS Request should be sent to the other node to inform it that it is able to communicate. This status should only be used in a D-STATUS Request or Response PDU.

7.6.1.2.11 No error

This value in the status field indicates that no errors exist at the present time and all layers are available. This status should be sent in a D-ACK PDU when using DATA/ACK service class or in a D-STATUS Request or Response PDU. It is advisable to send this status after an error condition has been resolved.

7.6.2 FRAME CHECK SEQUENCE

The Frame Check Sequence (FCS) is used to test the validity of the received data. The FCS is commonly referred to as the check sum. This word will be the one"s complement of the truncated binary sum of all the words received in the frame excluding the FCS word itself.
This Part of the Standard only requires error detection.

7.6.3 D-DATA REQUEST PDU

The D-DATA Request PDU is passed between peers to transport an STN Layer packet. The D-DATA Request PDU is not acknowledged by the peer. Table 7.6-5 defines the parameters required for this PDU.

Table 7.6-5—D-DATA request PDU fields

Item words	Field Name	Description
1	Frame Descriptor Word	The Frame Descriptor Word contains the D-DATA Request Code from Table 7.6-3 and the number of words in the STN packet.
2-n	Data	STN packet
n+1	Frame Check Sequence	The Frame Check Sequence for this frame.

7.6.4 D-DATA_ACK REQUEST PDU

The D-DATA_ACK Request PDU is passed between peers to transport an STN Layer packet. The D-DATA_ACK Request PDU is acknowledged by the peer with either a D-ACK or D-NACK PDU. Table 7.6-6 defines the required parameters for the D-DATA_ACK PDU.

Table 7.6-6—D-DATA_ACK request PDU fields

Item words	Field Name	Description
1	Frame Descriptor Word	The Frame Descriptor Word contains the D-DATA_ACK Request Code from Table 7.6-3 and the number of words in the STN packet.
2-n	Data	STN packet
n+1	Frame Check Sequence	The Frame Check Sequence for this frame.

7.6.5 D-ACK PDU

The D-ACK PDU is passed between peer Data Link Layers to acknowledge successful reception of a D-DATA_ACK Request PDU. Table 7.6-7 defines the required parameters for the D-ACK PDU.

Table 7.6-7—D-ACK PDU fields

Item words	Field Name	Description
1	Frame Descriptor Word	The Frame Descriptor Word contains the D-ACK Code from Table 7.6-3 and the current status as defined in Table 7.6-4
2	Frame Check Sequence	The Frame Check Sequence for this frame.

7.6.6 D-NACK PDU

The D-NACK PDU is passed between peer Data Link Layers to acknowledge the erroneous reception of a D-DATA_ACK Request PDU. The D-NACK PDU must return its current status. This status shall contain, among other possible status codes, the error encountered in the just received D-DATA_ACK PDU. Table 7.6-8 defines the parameters required for the D-NACK PDU.

Table 7.6-8 D-NACK PDU fields

Item words	Field Name	Description
1	Frame Descriptor Word	The Frame Descriptor Word contains the D-NACK Code from Table 7.6-3 and the current status as defined in Table 7.6-4
2	Frame Check Sequence	The Frame Check Sequence for this frame.

7.6.7 D-STATUS REQUEST PDU

The D-STATUS Request PDU is used to request the current status of the peer Data Link Layer and the on-line status of the upper layer"s protocol stack. The current local status shall be returned in a D-STATUS Response PDU. Table 7.6-9 defines the parameters required for the D-STATUS Request PDU.

Table 7.6-9—D-STATUS request PDU fields

Item words	Field Name	Description
1	Frame Descriptor Word	The Frame Descriptor Word contains the D-STATUS Request Code from Table 7.6-3 and the current status as defined in Table 7.6-4.
2	Frame Check Sequence	The Frame Check Sequence for this frame.

7.6.8 D-STATUS RESPONSE PDU

The D-STATUS Response PDU is passed between peers in response to a D-STATUS Request PDU. The current local status shall be sent in the D-STATUS Response PDU. Table 7.6-10 defines the parameters required for this PDU.

Table 7.6-10—D-STATUS response PDU fields

Item words	Field Name	Description
1	Frame Descriptor Word	The Frame Descriptor Word contains the D-STATUS Response Code from Table 7.6-3 and the current status as defined in Table 7.6-4.
2	Frame Check Sequence	The Frame Check Sequence for this frame.

7.6.9 D-ECHO REQUEST PDU

The D-ECHO Request PDU is passed between peers to transport a know data packet that is to be returned by the peer. The D-ECHO Request PDU is responded to by the peer with a D-ECHO Response PDU. Table 7.6-11 defines the required parameters for the D-ECHO Request PDU.

7.6.10 D-ECHO RESPONSE PDU

The D-ECHO Response PDU is passed to the peer in order to return the data received in a D-ECHO Request PDU. Table 7.6-12 defines the parameters required for this PDU.

Table 7.6-11—D-ECHO request PDU fields

Item words	Field Name	Description
1	Frame Descriptor Word	The Frame Descriptor Word contains the D-ECHO Request Code from Table 7.6-3 and the number of words in the data field.
2-n	Data	0-2048 words of data to be returned by the peer.
n+1	Frame Check Sequence	The Frame Check Sequence for this frame.

Table 7.6-12 D-ECHO response PDU fields

Item words	Field Name	Description
1	Frame Descriptor Word	The Frame Descriptor Word contains the D-ECHO Response Code from Table 7.6-3 and the number of words in the data field.
2-n	Data	0-2048 words of data received in the D-ECHO Request PDU.
n+1	Frame Check Sequence	The Frame Check Sequence for this frame.

8 Point to point physical layer protocol

8.1 *Introduction*

The point to point Physical Layer protocol defines the interface used across the plane between two devices in order to support high speed transfers of blocked data. Some considerations taken when defining this interface include the following:

— be simple enough to be retrofitted into existing equipment

— use a single connector of readily available style

— carry 16-bit parallel data with parity

— use differential circuits to ease ground and noise problems

— use circuitry to allow data signaling rate of at least 8 Mbytes per second

— use asynchronous "handshake" word transfer protocol so devices of varying instrinsic speed can be supported

— use identical interface symmetry so any two devices can be connected as equals or peers (i.e., no master-slave relationship).

The hardware defined for this interface provides signal ground and 24 differential circuits: 16 for data, 1 for parity, 6 for control, and 1 reserved for future use.

The target data rate is 8 Megabytes/second. The hardware interface may be capable of successfully operating at faster rates than the target rate. Asynchronous handshake protocols are used, thereby permitting devices to successfully connect to this interface and operate at slower data rates.

8.2 *Physical layer electrical specifications*

8.2.1 DEFINITIONS

The terms asserted, released and non-asserted are used to signify the state of a given line. For equivalence to Logical, Differential and EIA 485 specification, see Table 8.2-1.

A plus (+) following a signal name indicates the non-inverting element of a differential pair and the "A" terminal from the EIA 485 specification.

A minus (-) following a signal name indicates the inverting element of a differential pair and the "B" terminal from the EIA 485 specification.

The initial state is defined as having all bi-directional signals in the released state and Request Out (REQO), Interrupt Out (INTO), and Strobe Out (STBO) non-asserted.

Table 8.2-1—Signal state definitions

State	Logical	Differential	EIA485
Asserted	True	Positive	On
Non-asserted	False	Negative	Off
Released	Don't Care	Not Defined	Passive

8.2.2 SIGNAL DEFINITIONS

The physical DICOM interface and the logical interface plane between two connected devices is shown in Figure 8.2-1. The three pairs of control signals are crossed so that any device handles the signals in the same manner. The signals, their function, and protocol are defined in Tables 8.2-1 and 8.2-2. The parity signal shall be asserted if and only if an even number of data signal lines are asserted, thus providing odd parity.

Table 8.2-1—Electrical connections

PIN	Signal Name	PIN	Signal Name
1	Signal Ground	26	Signal Ground
2	D0+	27	D0-
3	D1+	28	D1-
4	D2+	29	D2-
5	D3+	30	D3-
6	D4+	31	D4-
7	D5+	32	D5-
8	D6+	33	D6-
9	D7+	34	D7-
10	D8+	35	D8-
11	D9+	36	D9-
12	D10+	37	D10-
13	D11+	38	D11-
14	D12+	39	D12-
15	D13+	40	D13-
16	D14+	41	D14-
17	D15+	42	D15-
18	PAR+	43	PAR-
19	Reserved	44	Reserved
20	STBO+	45	STBO-
21	STBI+	46	STBI-
22	REQO+	47	REQO-
23	REQI+	48	REQI-
24	INTO+	49	INTO-
25	INTI+	50	INTI-

Table 8.2-2—Electrical signal function

Signal Name	Function
Signal Ground	Signal return path between the circuit grounds
D0 to D15	Bi-directional data. D0 will carry the least significant bit of a transferred word.
PAR	Parity. Bi-directional signal that will be asserted if the number of asserted signals in D0 to D15 is even, thus producing odd parity.
STBO	Data Strobe Out. Output signal driven by the equipment interface. When transmitting data, the asserted state will signify that the data lines are stable and the receiver can now accept the data. When receiving data, assertion will signify that the data have been accepted.
STBI	Data Strobe In. A flow control input signal that is connected to the STBO signal from the equipment across the interface as seen in Figure 2. When transmitting data, detecting assertion of this signal means the data is accepted. When receiving data, assertion means new data is available.
REQO	Request to Send Out. An asynchronous output signal asserted by the equipment requesting to send data. The requestor must wait the arbitration time of one microsecond before assuming control. If REQI is asserted at the end of the arbitration wait time, a collision has occurred. REQO must be non-asserted for a collision wait time specified in 8.3.2.
REQI	Request to Send In. An asynchronous input signal connected to the REQO of the equipment on the other side of the interface. Detecting the assertion of this signal signifies that equipment wants to send data. If detected at the end of the arbitration wait time after asserting REQO, a collision has occurred.
INTO	Interrupt Out. An asynchronous output signal used to abort data transfer. It shall be asserted for one millisecond, then non-asserted. STBO and REQO shall be non-asserted during INTO assertion.
INTI	Interrupt In. An asynchronous input signal that is connected to INTO of the equipment on the other side of the interface. Detecting the assertion of this signal signifies the interruption of the data transfer. INTO, REQO and STBO shall be non-asserted.

8.2.3 SIGNAL ELECTRICAL CHARACTERISTICS

8.2.3.1 Driver and receiver characteristics

The electrical characteristics of the signals are specified in EIA 485, which allows for multiple generators and receivers to be attached to a common interconnecting cable. The specifications of the generators are structured so that more than one generator driving a line at once will not cause damage. Application of EIA 485, therefore, allows implementation of balanced bi-directional data and parity signals across the DICOM interface plane.

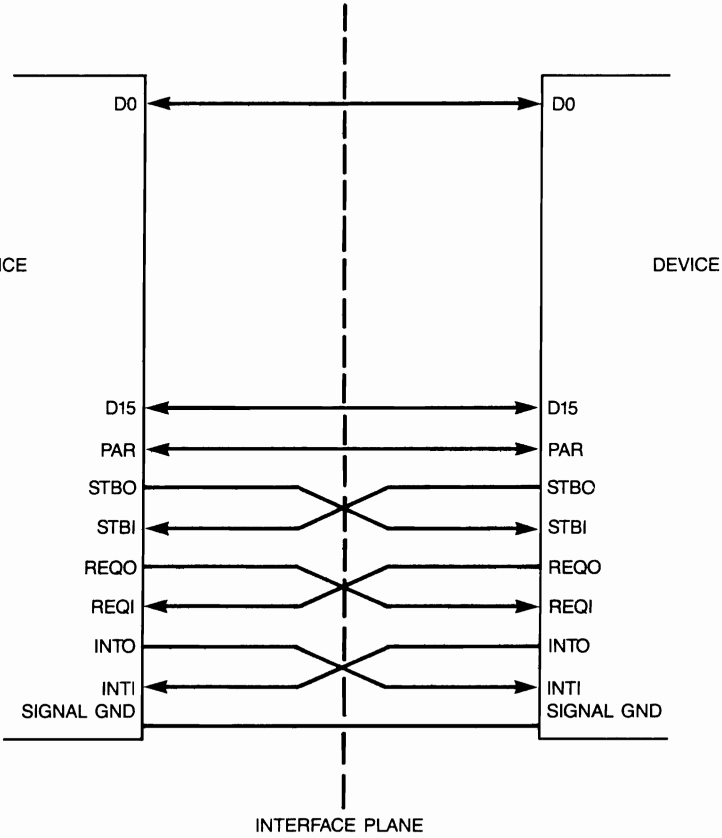


Figure 8.2-1—Physical and logical interface

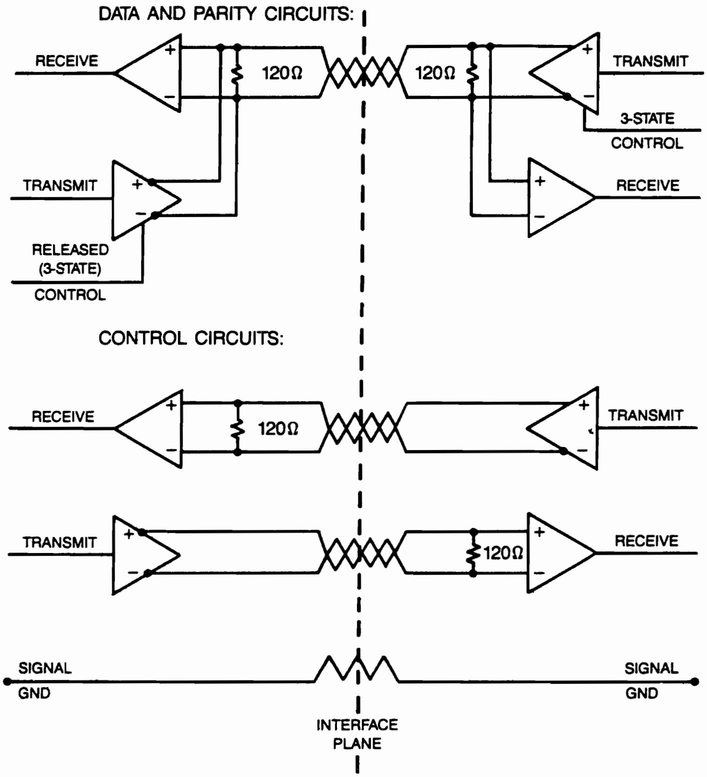


Figure 8.2-2—Typical interface circuit

8.2.3.2 Cabling and termination

Each circuit will have a 120 ohm termination resistor connected between the + and - inputs of the receiver. These resistors shall be located as close as possible to the receiver input terminals. The cables shall have a nominal impedance of 120 ohms.

The cable must be terminated with a 50 pin male micro-type connector. The six control circuits are crossed in the cable between ends so that the control lines are each driven by one side of the interface. It is recommended that a screw type retention mechanism be used. The studs for the connection should appear on the equipment connector and the screws on the cable connector.

The EIA 485 specification further describes cable and ground considerations. Figure 8.2-1 illustrates the crossing of the control circuits in the cable. Figure 8.2-3 shows typical circuits.
The connector on the equipment shall be a 50 pin female micro-type receptacle. Tables 8.2-1 and 8.2-2 show the pin definitions for the interface circuits. Figure 8.2-3 shows the mechanical characteristics of the connector.

8.2.3.4 Signal timing specifications

The timing to be used for sending data is shown in Figure 8.2-4, where time intervals are defined and minimum and maximum times are specified. These times define the handshake cycle of the STBO and STBI circuits that provide an unambiguous sequence of states for each word transfer. A minimum data circuit set-up time of 70 nanoseconds before assertion of STBO is specified. The maximum time of 1 second defines a physical layer "inactivity" time-out, beyond which successful transmission is not assured.

8.2.3.5 Error rates

The sum of the individual error rates on the 17 parallel lines in the interface shall be no greater than 17 x 10- ^{9}, prior to any correction. Error correction algorithms are not specified in the point to point standard.

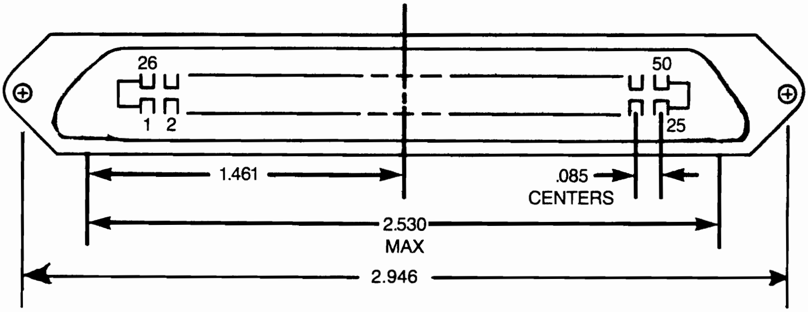
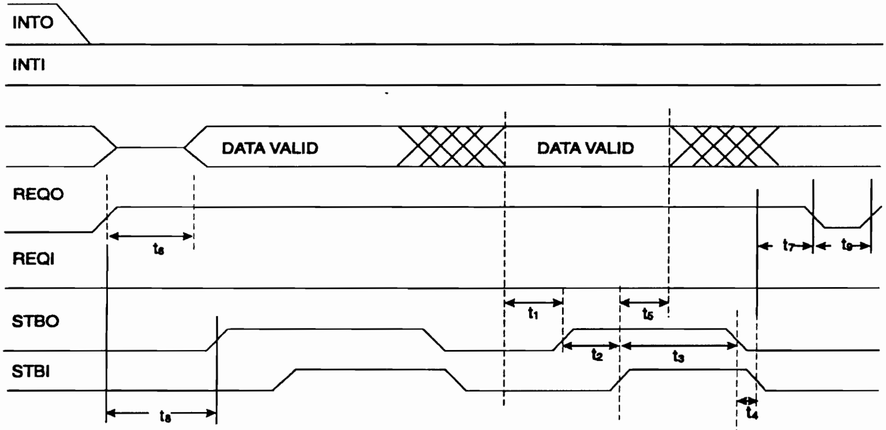


Figure 8.2-3—50 pin female receptacle front view



REF	PARAMETER	TIMING MIN	MAX
t1	DATA SET UP TO LEADING EDGE OF STBO	70 nanoseconds	1 second
t2	LEADING EDGE OF STBO TO LEADING EDGE OF STBI	0 nanoseconds	1 second
t3	LEADING EDGE OF STBI TO TRAILING EDGE OF STBO	0 nanoseconds	1 second
t4	TRAILING EDGE OF STBO TO TRAILING EDGE OF STBI	0 nanoseconds	1 second
t5	LEADING EDGE OF STBI TO DATA CHANGING	0 nanoseconds	
t6	COLLISION ARBITRATION WINDOW	750 nanoseconds	1250 nanoseconds
t7	TRAILING EDGE OF STBI TO TRAILING EDGE OF REQO	0 nanoseconds	1 second
t8	LEADING EDGE OF REQO TO LEADING EDGE OF STBO	(t6) ns + 70 nanoseconds	1 second
t9	TRAILING EDGE OF REQO TO LEADING EDGE OF REQO	10 microseconds	

Figure 8.2-4—Timing diagram for sending data

8.2.3.6 Rate of data transfer

Using currently available logic devices, the interface circuitry should allow data signaling rates in excess of 8 Mbytes per second. Cable propagation delay and data handling in the host equipment will constrain the actual interface data rate. There is no requirement to meet a specific data rate across the physical interface other than the 2 byte per second rate implied by the physical layer time-out, described in section 8.2.3.4.

Annex A specifies how to determine maximum transfer rate.

8.3 *Physical layer protocol*

8.3.1 INTRODUCTION

The physical interface consists of sixteen bi-directional data lines plus one bi-directional parity line, and six unidirectional control lines as described in 8.2. These control circuits provide for interface arbitration and data flow control and provide hardware interrupt service. The Physical Layer data transfer protocol is described in terms of a transition sequence state model that uses the control circuit states to specify the protocol states (i.e., the transition labels from state to state are not events that a state machine must sense to move from state to state). The states as defined by the control circuits refer to the states of these lines at the interface connector to a cable, and not to states in a state machine implementation of the protocol. Thus the state sequence could be verified by a logic analyzer attached to an interface connector.

8.3.2 PHYSICAL LAYER STATE MODEL

The Physical Layer states are defined in Table 8.3-1 and Figure 8.3-1 shows the diagram of allowed state transition sequences. The binary state values assigned to the states represent the states of the control signals in the order INTO, INTI, REQO, REQI, STBO, STBI. The values for INTI, REQI, and STBI are the signal values seen by the device receivers. The values for INTO, REQO, and STBO are the signal values output. These values are either asserted (1) or non-asserted (0). For example, RX-INT with a value of 0110XX means INTO is non-asserted, INTI and REQO are asserted, REQI is non-asserted, and STBO, STBI are in either state. The state diagram shown in Figure 8.3-1 defines all functions of the interface since the circuitry on both sides of the interface is functionally equivalent. The initial state is the IDLE state and is defined as having all bi-directional signals in the released state and REQO, INTO, and STBO non-asserted.

The state diagram for signals at the connector should be interpreted with reference to the timing diagram of Figure 8.2-4. Both timing and control signal states are referenced to the interface connector. Signal delays from long connecting cables will cause slower transversal of protocol state sequences. The timing diagram specifies a minimum delay of ten microseconds between the non-assertion of REQO by a sender and the re-assertion of REQO for a next frame to allow reaction and arbitration time for the receiver. The receiver need not wait this interval to arbitrate for the interface.

The RELEASE state of Figure 8.3-1 is a transition state in which one device has released its control signals and the other side has not. The second device will pass directly to IDLE when its control signals are released.

The arbitration for permission to send data is described by the state diagram (Figure 8.3-1). The bi-directional data and parity signals (D0-D15 and PAR) must be left in the released state until permission is granted to send data as defined by entering the TX_READY state. When a collision occurs (entering the

COLLISION state), the device detecting the collision must delay for a collision wait time before trying to gain control of the interface again. This collision wait time shall be a uniformly distributed random time between 0 and 15 microseconds or an implementation-specific fixed time greater than 7.5 microseconds. During the delay, the interface shall pass through the RX_READY state before data is to be received.

A receiver must be prepared to receive a zero length frame. Such a frame has no meaning to the receiver and must be ignored. A zero length frame occurs naturally if both peers de-assert their REQO signals as a result of collision during interface arbitration.

8.3.3 EXCEPTION HANDLING

The Physical Layer exception handling mechanism relies on the INTO interrupt signal. The protocol specifies that the interrupted equipment will respond by sending a Data Link Layer Status Request frame. Section 7.2. describes this mechanism in more detail.

The Physical Layer will assert the interrupt signal (INTO) if the physical state sequence protocol is violated. This action will be reported to the interface's Data Link Layer. The Data Link Layer will attempt to report this error to the peer as a Physical Layer Protocol Error (see section 7.2.2.). Three Physical Layer Protocol Error situations are defined:

- False Collision
- Physical State Sequence Violation
- Physical Layer Time-out

The Physical Layer also can be instructed by upper layers to assert it"s INTO signal. This mechanism is used for Data Link Layer data class flow control. This mechanism is described in Section 7.2.
The Physical Layer, upon detection of an INTI assertion, will report the assertion to it"s Data Link Layer for resolution. The Physical Layer will also de-assert it"s REQO, STBO and INTO signals.

8.3.4 FLOW CONTROL

The only flow control mechanism at the Physical Layer for interface access and word transfer is the interlocked handshake cycle of STBO and STBI.

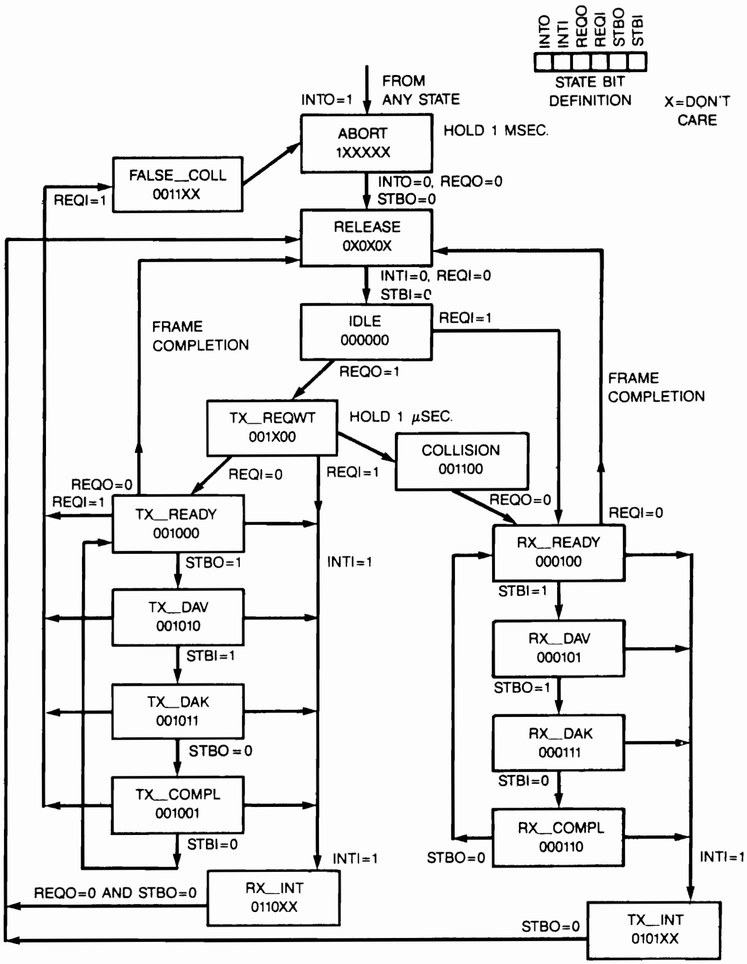


Figure 8.3-1—Physical layer state diagram

Table 8.3-1—Description of physical protocol states

State Name	State Code	Description
IDLE	000000	The IDLE state is the normal quiescent state of no data transfer. PAR and Data signals are in the RELEASED state.
RX_READY	000100	The peer equipment is asserting its REQO resulting in REQI being asserted. Prepare to receive data.
RX_DAV	000101	Data are available on the data circuits, as signaled by STBI asserted.
RX_DAK	000111	STBO is asserted to acknowledge receipt of data. The current transfer is complete, and the data shall be changed, if desired.
RX_COMPL	000110	STBI non-asserted. Data are no longer valid. Non-assert STBO to return to RX_READY.
TX_REQWT	001X00	REQO is asserted from IDLE state to request permission to send, but not during collision wait time. Hold for 1 microsecond.
COLLISION	001100	REQO and REQI asserted, resulting in a collision for interface transmission rights. De-assert REQO, start collision wait timer, enter RX_READY state.
TX_READY	001000	Interface is ready for transmission of data. Place next word on data circuits.
TX_DAV	001010	Assert STBO to signal data are stable at least 70 nanoseconds after asserting data on D0-D15 and PAR.
TX_DAK	001011	STBI asserted, acknowledge receipt of data.
TX_COMPL	001001	Non-assert STBO to signal data no longer stable. Prepare next word for transmission. Return to TX_READY when STBI is non-asserted.
RX_INT	0110XX	Peer device has interrupted the data transfer. Deassert STBO, REQO.
TX_INT	0101XX	Peer device has interrupted the data transfer. Deassert STBO, REQO.
ABORT	10XXXX	Assert INTO to abort data transfer. This shall be done if an undefined or out-of-sequence state is detected that interferes with the specified protocol for data transfer. Hold for 1 millisecond.
RELEASE	0X0X0X	INTO, REQO, STBO are de-asserted. Any INTI, REQI, STBI can be asserted. This state is a transition state to IDLE. The device can assert INTO after 1 second if these circuits remain asserted and are preventing desired activity.
FALSE_COLL	0011XX	REQI is asserted during transmission cycle. This is a false collision error state.

9 Conformance

9.1 *Conformance requirements*

An implementation claiming conformance to DICOM V3.0 Point to Point Communication Support shall:

a) Meet the Session/Transport/Network Layer Protocol Requirements as defined in section 6 of this Part. This requires specifying the following implemented options and values:

- supported STN Service Class (DATA/ACK or both DATA and DATA/ACK)
- implemented timers and their respective time-out values (S-CLOSE timer, release timer, etc)
- actions taken when timers expire

b) Meet the Datalink Layer Protocol Requirements as defined in section 7 of this Part. This required specifying the following implemented options and values:

- retry counts and actions taken when the retries are exhausted
- number of channels supported (including Channel 0)
- Channel Number limitations (ie, the largest value Channel Number can contain)

c) Meets the Physical Layer Electrical, Mechanical and Protocol Requirements as defined in section 8 of this Part.

9.2 *Conformance claim*

An implementation claiming conformance to DICOM V3.0 Point to Point Communication Support shall state:

"DICOM V3.0 Point to Point Communication Support with the following relevant implementation information." This implies that the conformance requirements defined in section 9.1 of this Part are met and the options are identified.

Annex A
(Normative)
Maximum Transfer Rate Measurement

A.1 Test circles

The test circuits shown in Figures A-1 and A-2 can be used to measure the maximum transfer rate of the DICOM interface. The circuit should be constructed as close as possible to its male connector to minimize circuit delays. The minus reference of the +5V supply used to power the pullup resistors R1 and R2 should be connected to the connector ground pins 1 or 26 as shown in Figures A-1 and A-2.

The capacitor C2 should have good high frequency characteristics.
The test circuit shown in Figure A-2 can be used to measure the transfer rate of a DICOM interface with an attached cable (see 7.2.). This test circuit should have a female connector and may be used to test transfer rates with various cable lengths.

A.2 Measurement

The test should be performed by instructing the interface under test to send frames of 2052 words across the interface. These frames should be sent as DATA service frames as the test circuit does not have the ability to acknowledge DATA/ACK service frames. It also should be noted that any frame type requiring a response such as STN-STATUS Request will not receive a response.

The data transfer rate should be computed by measuring the time t1, as shown in Figure A-3, with an oscilloscope or other suitable measuring instrument at test point TP1, as shown in Figures A-1 and A-2. Time t1, should be the time from the output of the first word of the first frame to the first word of the second frame. The number of bytes (4104) should be divided by the time t1 obtained in the previous step to obtain the maximum transfer rate in bytes/second.

As an example, if the time t1 measured using the above procedure was determined to be 5.1 x 10-4 seconds the maximum transfer rate would be computed as:

TRANSFER RATE = (4104 BYTES/ 5.1 x 10-4 SEC) = 7.039 x 106 BYTES/SEC

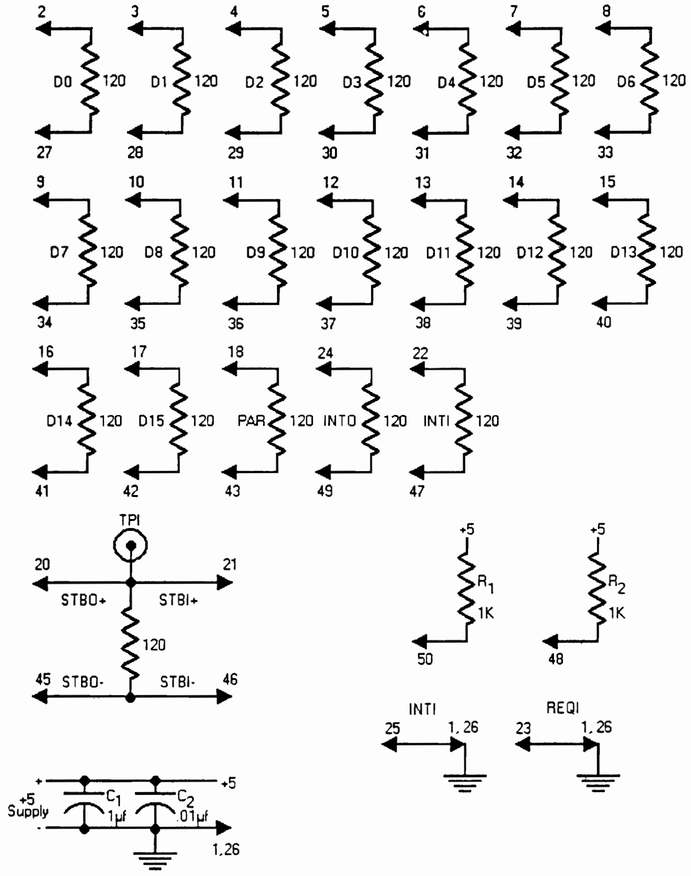


Figure A-1—Test circuit for no cable with male connector

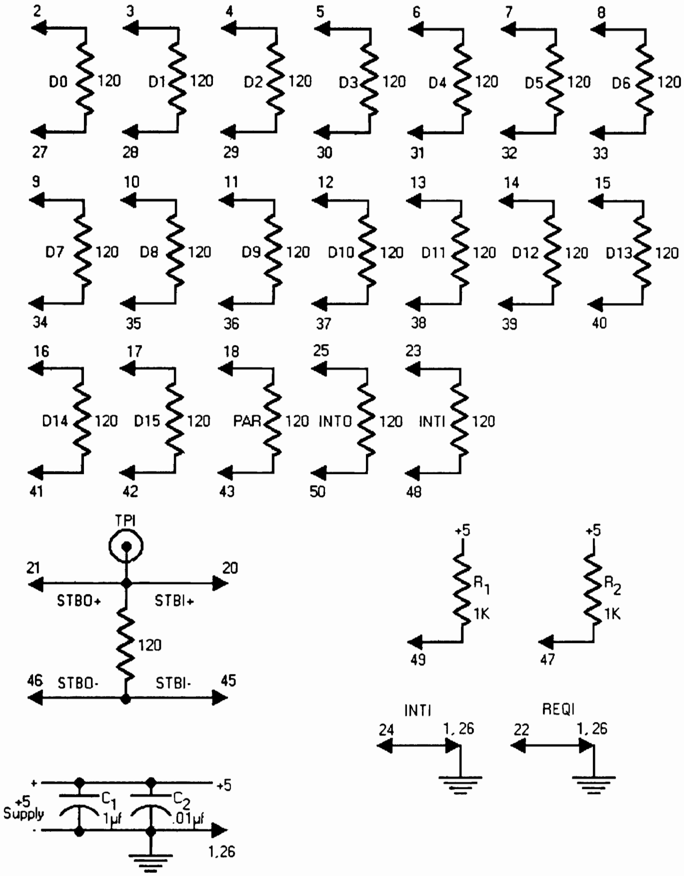


Figure A-2—Test circuit for cable with female connector

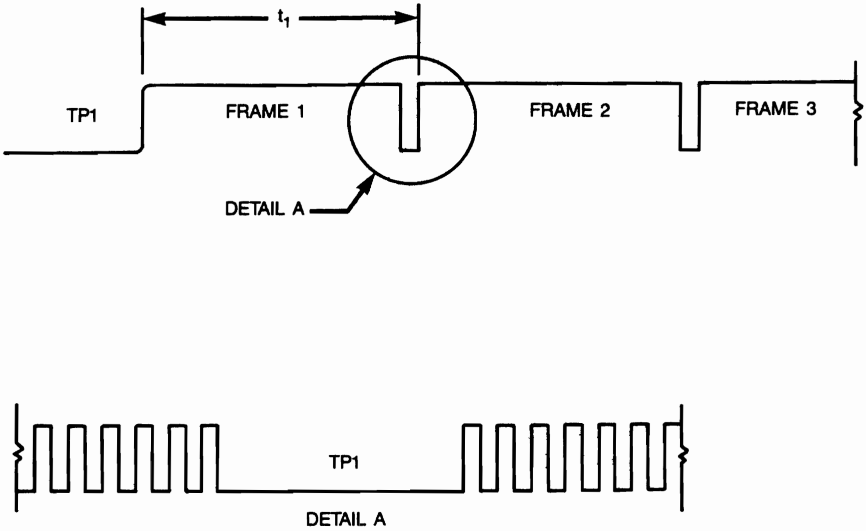


Figure A-3—Transfer time measurement

Annex B
(Normative)
Differences from the ACR-NEMA 300-1988 standard

This annex details the differences between the DICOM point to point interface and the ACR-NEMA 300-1988 standard interface. The differences are minor and should not effect interoperability of DICOM point to point interfaces and those of ACR-NEMA 300-1988. Most of the differences between the DICOM point to point interface and the ACR-NEMA 300-1988 standard interface are a direct result of the implementation experience of those vendors who participated in the ACR-NEMA Hardware Connectivity Workshop in May of 1990.

B.1 Session/transport/network layer differences

The DICOM point to point Session/Transport/Network Layer is little changed from the ACR-NEMA 300-1988 Session and Transport/Network Layers. Although the formalism presented in this document (ie, the Session/Transport/Network Layer) differs from that of the ACR-NEMA 300-1988 document, the peer to peer protocol is identical, with a few exceptions. These exceptions are as follows:

a) The reserved bits in the Connection Header Connection Control Word (ACR-NEMA 300-1988; Table 8-6) are still reserved. These bits, however, are now explicitly required to be zero until their use is defined. Further, the Continue Connection Control Word is removed from the standard. Only the Connect Request, Connect Response, Disconnect Request and Disconnect Response Connection Control Words are valid.

b) The Connection Header figure (Figure 6.8-2) appears different in this version of the Standard, but the header is actually the same as in ACR-NEMA 300-1988. The figure was redrawn to eliminate confusion concerning the byte ordering of the Application and Network addresses.

c) Default Source and Destination addresses have been omitted from the DICOM Standard.

d) The operation of the Session/Transport/Network Layer Pause/Resume mechanism was unclear in the ACR-NEMA 300-1988 standard. This was a result of Figure 8-6 in the old standard document. The committee"s intention has always been that the Pause/Resume mechanism work as described in section 6.5 in this Part of the DICOM Standard.

e) The reserved bits in the Service Class parameters in the Packet Descriptor Word (ACR-NEMA 300-1988; Table 8-5) are still reserved. These bits, however, are now explicitly required to be zero until their use is defined. Default service class is identical to DATA/ACK service class (section 6.8.1.1.3). Support of the DATA service class is now optional.

f) The Reset Channel Request (RCR) and Reset Channel Indication (RCI) packet types have been removed from the standard. The use of these packet types resulted in much confusion and have been deemed to be of no real value. All known ACR-NEMA 300-1988 implementations handle these packet types as "unknown packet type," ie., the receiver of such a packet closes the associated channel. This is identical to the treatment of an "unknown packet type" as specified in this Part of the DICOM Standard.

These differences have no impact on interoperability of ACR-NEMA 300-1988 and DICOM point to point interfaces. There are no known implementations of ACR-NEMA 300-1988 which either made use of the

Continue Connection Control Word or set the reserved bits in the Connection Control Word or the Service Class to a value other then zero. The default Destination/Source Field inconsistency has been addressed by all known implementors.

B.2 Data link layer differences

The DICOM point to point Data Link Layer is little changed from the ACR-NEMA 300-1988 Data Link Layer. Although the Data Link Layer formalism presented in this document differs from that of the ACR-NEMA 300-1988 document, the peer to peer protocol is identical with a few exceptions. The differences with and corrections to the ACR-NEMA 300-1988 standard are as follows:

a) The frame status and error bits are defined and their use described in more detail in this document than in the ACR-NEMA 300-1988 document. The transport of these bits across the peer to peer interface is now understood to have no effect on protocol recovery. The issue of exception handling at the Data Link Layer is more fully addressed in this document.

b) The Data Link protocol state diagram (Figure 8-3; ACR-NEMA 300-1988) has been changed slightly. This change only effects the usage of the ACK and NACK frames. These frames are now allowed to occur with a STATUS REQ/STATUS IND exchange occurring between the acknowledgement frame and the frame being acknowledged. That is to say, the STATUS REQ/STATUS IND exchange is seen to be orthogonal to the "normal" Data Link protocol state diagram.

These differences may slightly impact the interoperability of ACR-NEMA 300-1988 and DICOM point to point interfaces. This situation, however, is no worse than the existing interoperability situation of ACR-NEMA 300-1988 interfaces of different manufacturers. These changes to the ACR-NEMA 300-1988 Data Link Layer specification are seen to be required based on the interoperability experience gained by implementors at the ACR-NEMA Hardware Connectivity Workshop held in May of 1990.

B.3 Physical layer differences

There are no differences between the DICOM point to point interface standard and the ACR-NEMA 300-1988 standard in regards to the Physical Layer. A number of typographic errors and inconsistencies in ACR-NEMA 300-1988 tables have been corrected.

Annex C
(Informative)
Exception Handling

Exception handling can be performed by using the D-STATUS PDU or by interrupts as described in section 7.2 of this Part. This Annex will show examples of handling some of the exceptions that can occur over a point to point link. Note that these examples do not represent all possible exceptions nor does it imply that the method shown is always the best one.

Parity errors, frame check sequence errors and incomplete frames are all conditions that are associated with frame transfer errors. The resulting action should be to attempt a limited number of retries to retransmit the frame. If the error is resolved during these retransmissions, the next frame is transmitted. If not, a D-STATUS Request PDU with a status of Retry Count Exceeded is composed and delivered.

Transmitter		**Receiver**
D-DATA_ACK (BSN=20,	---->	
CHN=1)	<----	
		D-NACK (Frame Check Sequence Error)
D-DATA_ACK (BSN=20,	---->	
CHN=1)	<----	
		D-NACK (Frame Check Sequence Error)
D-DATA_ACK (BSN=20, CHN=1)	---->	
	<----	D-NACK (Parity Error)
D-STATUS Request (Retry Count Exceeded)	---->	
	<----	D-STATUS Indication (current status of interface)

Instead of sending a D-NACK PDU when the previous frame type is unrecognized, a D-STATUS PDU with the status of Unrecognized Frame could be issued. The reason for using the D-STATUS PDU is that the frame type was unknown and the D-NACK is only issued when the peer is aware that the frame type was DATA/ACK. This is illustrated by the following examples.

Transmitter		**Receiver**
D-DATA_ACK (BSN=13,	---->	
CHN=1)	<----	
		D-STATUS Request (Unrecognized Frame)
D-STATUS Indication	---->	
(No Error)	---->	
	<----	D-ACK (No Error)

The same consideration must be given for a D-STATUS Request PDU. If a D-STATUS Request PDU is unrecognized, it will be followed by a D-STATUS Request with a Frame Unrecognized status. This is the only case where a D-STATUS Request PDU may follow a D-STATUS Request PDU.

Transmitter		**Receiver**
D-DATA_ACK (BSN=13, CHN=1)	---->	
D-STATUS Request (Acknowledge Pending)	---->	
	<----	D-STATUS (Unrecognized Frame)
D-STATUS Indication (No Error)	---->	
D-DATA_ACK (BSN=13, CHN=1)	---->	
	<----	D-ACK (No Error)

When receiving Unavailable Receive Buffer status, the data transmitter needs to know if it should retransmit the last frame. The following example shows that the receiver has already received the frame.

Transmitter		**Receiver**
D-DATA_ACK (BSN=70, CHN=2)	---->	
D-STATUS Request (Acknowledge Pending)	---->	
	<----	D-STATUS Indication (Unavailable Receive Buffer)
	<----	D-ACK (No Error)
D-DATA_ACK (BSN=71, CHN=2)	---->	

In the following example, the Unavailable Receive Buffer status indicates that the receiver has already received the frame but it cannot accept more at this time. The transmitter queries the receiver until a buffer is available.

Transmitter		**Receiver**
D-DATA_ACK (BSN=70, CHN=2)	---->	
	<----	D-NACK (Unavailable Receive Buffer)
D-DATA Request (No Error)	---->	
	<----	D-STATUS (Unavailable Receive Buffer)
D-STATUS Request (No Error)	---->	
	<----	D-STATUS Indication (No Error)
D-DATA_ACK (BSN=71, CHN=2)	---->	

Receiving a D-NACK PDU with an Unavailable Receive Buffer status implies that the frame just received was not accepted and the last frame should be retransmitted. Note that before resending the last frame, a D-STATUS Request PDU is sent. If the exception has been resolved, the last frame is retransmitted.

Transmitter		**Receiver**
D-DATA_ACK (BSN=70, CHN=2)	---->	
	<----	D-NACK (Unavailable Receive Buffer)
D-STATUS Request (No Error)	---->	
	<----	D-STATUS Indication (No Error)
D-DATA_ACK (BSN=70, CHN=2)	---->	
	<----	D-ACK (No Error)

Upon running out of buffer space, the receiver could also interrupt the sender and respond to a D-STATUS Request from the sender with an Unavailable Receive Buffer status. In this case the last frame should be retransmitted.

An Unspecified Error status may have been originated from the errors presented in the following examples. Upon receiving such a status, the sender should retransmit the last frame.

Transmitter		Receiver
D-DATA_ACK (BSN=30, CHN=1)	---->	
	<----	D-NACK_ACK (BSN=42, CHN=4)
D-STATUS Request (Unspecified Error)	---->	
	<----	D-STATUS Indication (No Error)
D-DATA_ACK (BSN=30, CHN=1)	---->	
	<----	D-ACK (No Error)

The next example shows use of the interrupts together with the D-STATUS Request PDU to determine the error. In the example, the transmitter has detected a violation to the Data Link protocol. The Physical Layer reports this error to the Data Link Layer for inclusion in the D-Status Request status field.

Transmitter		Receiver
INTO	---->	INTI
	<----	D-STATUS Request (No Error)
D-STATUS Request (Unspecified Error)	---->	

The following conditions might result in the initiation of an interrupt. If this occurs, the last unacknowledged complete frame sent across the interface should be retransmitted.

a) A time specified in Figure 8.2-4 exceeded the maximum value of 1 second (Physical Layer Time-out)

b) The Physical Layer state sequence protocol is violated (Figure 8.3-1) (Unspecified Error)

c) Applying flow control and/or exception handling to DATA or DATA/ACK service class

d) A hardware reset has occurred (Interface Not Serviced)

NATIONAL ELECTRICAL MANUAFACTURERS ASSOCIATION
1300 North 17th Street, Suite 1847, Rosslyn, Virginia 22209