

Data Interchange plc

Communication Standards

Issued: 14 March 2006

Copyright Data Interchange Plc
Peterborough, England, September 2005.

All rights reserved. No part of this document may be disclosed to third parties or reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written permission of Data Interchange Plc.

Table of Contents

Communication Standards	5
Introduction	5
The OSI Reference Model	5
<i>Introduction</i>	5
<i>Layer Description</i>	6
<i>Worked example</i>	8
<i>The seven layers</i>	8
X.25	10
<i>Overview</i>	10
<i>X.25 benefits</i>	11
<i>X.25 performance</i>	11
<i>X.25 and the OSI 7-layer reference model</i>	11
<i>X.25 support in DI products</i>	12
ISDN	12
<i>Overview</i>	12
<i>Packet Switching</i>	13
<i>ISDN support in DI products</i>	13
CAPI	13
<i>Overview</i>	13
<i>CAPI Messages</i>	14
<i>Message Queues</i>	15
<i>Utilising Message Queues</i>	15
<i>CAPI Configuration</i>	17
<i>RCAPI</i> 17	
<i>CAPI support in DI products</i>	18
TCP/IP	18
<i>Overview</i>	18
<i>TCP/IP benefits</i>	18
<i>TCP/IP and the OSI 7-layer reference model</i>	18
<i>IP addresses</i>	19
<i>TCP/IP support in DI products</i>	19
XOT	19
<i>Overview</i>	19
<i>XOT scenarios</i>	20
<i>XOT support in DI products</i>	22
PAD	22
X.28	22
<i>X.28 support in DI products</i>	23
X.29	23
X.3	23
X.31	23
<i>X. 31 support in DI products</i>	24
X.32	24
X.400	25
<i>X.400 support in DI products</i>	26

Communication Standards

Introduction

The deciding factor in choosing a connection method for communications is usually a question of the trading partner's requirements. Automotive manufacturers, major retailers, BACS and the Inland Revenue, for example, will dictate which method of communication is to be used. The companies which exchange electronic documents with these bodies therefore have little choice available, unless those bodies allow connections via a VAN.

With a VAN acting as middleman, a company has far more choice, since the VAN can accept communications via one method and forward them via another.

The next consideration, then, is likely to be cost, speed and security of each method.

On the whole, X.25 is very secure, but is slow and very expensive. TCP/IP is cheap and fast but can lack security (unless a secure point-to-point dial-up connection is made or an encrypted VPN tunnel is used). X.25 over ISDN falls in the middle, as it is secure and reasonably cheap.

There are other methods of using X.25, whereby a standard modem connection is made to a local X.25 provider which then makes an X.25 connection on the sender's behalf. This is called X.28 or, where the modem connection is encrypted, X.32.

Connections can also be made to the X.25 network for free using the D channel of an ISDN line (the D channel usually just controls the two B channels which are used for data). This method, X.31, is currently not available in the UK, but is available in several European countries.

The following sections give more detail about each of these communication standards and related areas. Whichever means of communication is used, the connection method needs to be the same at both ends. To explain why this is so, we begin with a description of the OSI Reference Model.

The OSI Reference Model

Introduction

At all levels of communication carried out between the computer systems and network facilities of different trading partners, there is a need for some form of standardisation, allowing different systems to communicate with each other, no matter what hardware they are using.

This requirement was met by the joint collaboration of the ISO (International Standards Organisation) and the CCITT (Consultative Committee for International Telegraph and Telephone), which produced

a hierarchical model known as the OSI (Open Systems Interconnection) seven-layer model.

The seven layers are:

- Level 7 Application Layer
- Level 6 Presentation Layer
- Level 5 Session Layer
- Level 4 Transport Layer
- Level 3 Network Layer
- Level 2 Data Link Layer
- Level 1 Physical Layer

Each layer performs a set of self-contained functions which contribute to the communications process and so have been logically grouped in this way. The lower levels in the model are closely linked to the physical electronics involved in sending and receiving data. The higher levels are linked more closely to the front-end application program used by the operator either to enter the data or to interpret it once it is received.

Layer Description

The 7 layers of the model can be divided into two groups:

The network-dependent group covers the first three layers (1 – 3). These share the common feature of dealing with protocols associated with the data communications network that physically links two or more computer systems together.

The second group covers levels 4 to 7. These deal with the applications of a user's system. They provide protocols which allow two end-user application processes to interact with one another.

The Transport Layer provides the pivot point for the whole model. It acts as an interface between the two groups of layers building on the network services provided by the lower three layers and making these available to the upper three application layers.

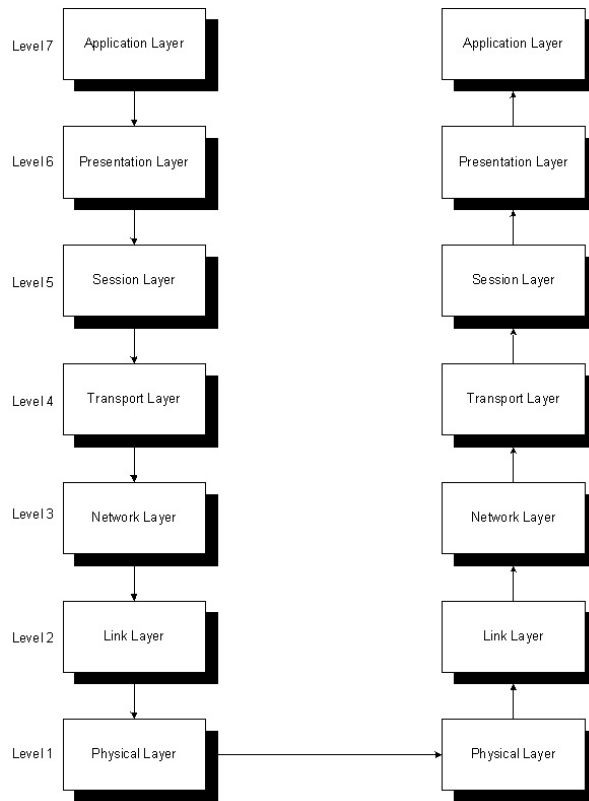


Figure 1 – the transfer of data across the OSI Model

1. User information creates the data (layers 5 – 7).
2. Data is converted to segments (layer 4)
3. Segments are converted to packets or datagrams (layer 3)
4. Packets, or datagrams, are converted to frames (layer 2)
5. Frames are converted to bits (layer 1)

Each level is defined by a protocol so that it is compatible with the corresponding layer (its peer layer) in another remote system with which it communicates.

Conceptually, two corresponding layers of two separate systems are able to communicate because they share the same protocol. In reality, each layer does not communicate directly with the same layer of another system. It relies on the level below it and provides services for the one above.

The result is that data is actually passed from the applications layer of one system down through the other levels to the physical layer where it is sent to the physical level of the intended receiver. The data is then passed up through the hierarchy of layers to the application level of the receiver's system. This is illustrated by the diagram above and explained in the section below.

Worked example

The information exchange process occurs between peer OSI layers. Each layer in the source system adds control information to data, and each layer in the destination system analyses and removes the control information from that data.

If System A has data from a software application to send to System B, the data is passed to the application layer. The application layer in System A then communicates any control information required by the application layer in System B by prefixing a header to the data.

The resulting information unit (a header and the data) is passed to the presentation layer, which prefixes its own header containing control information intended for the presentation layer in System B. The information unit grows in size as each layer prefixes its own header (and, in some cases, a trailer) that contains control information to be used by its peer layer in System B.

At the physical layer, the entire information unit is placed onto the network medium. The physical layer in System B receives the information unit and passes it to the data link layer. The data link layer in System B then reads the control information contained in the header prefixed by the data link layer in System A. The header is then removed, and the remainder of the information unit is passed to the network layer.

Each layer performs the same actions: The layer reads the header from its peer layer, strips it off, and passes the remaining information unit to the next highest layer. After the application layer performs these actions, the data is passed to the recipient software application in System B, in exactly the form in which it was transmitted by the application in System A.

The seven layers

Each of the seven layers is described below.

Level 7 The Application Layer

The Application layer is the layer closest to the end user. This layer interacts with software applications that implement a communications component.

This layer provides services such as the identification of the intended communication partners, determination of current availability of network resources and intended communications partner, establishing authority to communicate, agreement on privacy (encryption) mechanisms, and identification of constraints on data syntax.

Level 6 The Presentation Layer

The Presentation layer deals with data translation, compression and encryption. It's main purpose is to define data formats, such as ASCII, EBCDIC, binary and JPEG. For example, FTP allows you to specify a binary or ASCII transfer. If binary is chosen, the sender and receiver do not modify the contents of the file. If ASCII is chosen, the sender first translates the text into ASCII format before sending. On receipt, the receiver translates the text back from ASCII to the character set being used on the receiving computer.

Level 5 The Session Layer

The Session layer defines how to start, control and end electronic "conversations", known as sessions.

It's responsibilities include checkpointing, which allows the application to be notified if only part of a series of messages is successfully transmitted. For example, an ATM transaction, in which you take cash from your account, should not debit the account and fail before paying out the cash and then record the transaction even though you did not receive the money.

Level 4 The Transport Layer

The Transport layer accepts data from the Session layer and segments the data for transport across the network. It is responsible for ensuring the data is delivered without errors and in the correct sequence. This level also provides acknowledgements of successful transmissions.

At this level, data is held in information units known as segments.

Typical Transport layer functions include flow control, and error checking and recovery. Flow control manages data transmission between devices so that the transmitting device does not send more data than the receiving device can process. Error checking involves the creation of various mechanisms for detecting transmission errors, while error recovery involves taking actions, such as requesting that data be re-transmitted, to resolve any errors that occur.

Level 3 The Network Layer

The Network layer defines the network address of the destination system, which differs from the physical address defined in the Data Link layer.

It is responsible for addressing, determining routes for sending data packets (the data format at this level), and managing network problems such as packet switching, data congestion and routing.

Because this layer defines the logical network layout, routers can use this layer to determine how to forward data packets.

Level 2 The Data Link Layer

The Data Link layer provides reliable transit of data across a physical network link and defines the methods used to transmit and receive data on the network.

Different Data Link layer specifications define different network and protocol characteristics, including physical addressing, network topology, error notification, sequencing of frames (the data format at this level), and flow control.

Physical addressing (as opposed to network addressing) defines how devices are addressed at the Data Link layer. The Data Link layer also defines the network topology being used, which describes how devices are to be physically connected, such as in a bus or a ring topology.

Error notification alerts upper-layer protocols that a transmission error has occurred, and the sequencing of data frames re-orders frames that are transmitted out of sequence. Finally, flow control moderates the transmission of data so that the receiving device is not overwhelmed with more traffic than it can handle at one time.

Level 1 The Physical Layer

The Physical layer defines such things as cables, network interface cards and connectors, and characteristics such as voltage levels and data rates.

This level, where the data is in bit format, deals with the physical and electrical interface between the user's system and the network.

X.25

Overview

X.25 is a communications subsystem based on packet-switching technologies. Packet-switching technologies are protocols in which messages are divided into packets before they are sent. Each packet is then transmitted individually and can even follow different routes to its destination. Once all the packets forming a message arrive at the destination, they are recompiled into the original message.

Due to the original application of X.25 i.e. over potentially unreliable analog telephone lines, X.25 carries a significant error-checking overhead, as every device in the X.25 network acknowledges each packet it receives. This slows the transfer of information and uses up bandwidth, but does provide a highly reliable service.

To use native X.25, an X.25 communications adapter is required. This connects to your computer and enables it to communicate directly with a synchronous modem. The synchronous modem then drives a leased line to the packet switched exchange. The modem and leased line are

generally provided by your X.25 provider although the modem may be independently purchased.

X.25 benefits

Unlike asynchronous access, native X.25 access allows the packets of data to be transferred across the leased line and into the adapter connected directly to your machine. The packets of data are under the control of the HDLC error checking protocol, so the need for special logic extensions to check the validity of the data is eliminated, and data integrity is assured.

The second and probably more important benefit of native X.25 connection is that multiple simultaneous connections may be established across a single line. This means that you can be in communication with trading partners A, B and C and still have room left over for trading partner D to make an incoming call to you to send you urgent data. The number of simultaneous connections on each line will depend upon the line's characteristics as it was purchased from your X.25 service provider, but typical values are 4 sessions, 8 sessions, 32 sessions and above. The maximum theoretical limit is 255 sessions on a single line.

X.25 performance

Typical line speeds for a native X.25 connection would be 9600 baud up to 64K baud. What must be remembered when considering speed and number of sessions is that if the connection is 9600 baud and there are, for example, 4 concurrent sessions active, each session will be sharing the total bandwidth of 9600 baud. In other words, if all 4 are transmitting data each will get approximately 2400 baud worth of access time as the packets for each session are queued across the line. This must be remembered if a 9600 line is used with a very large number of circuits, say 30. On a busy system the data throughput rate may be quite drastically reduced if a large number of users start communicating at the same time. This type of scenario will lead to time-outs and other communications congestion problems. The alternative is to reduce the number of circuits on the line so that the line capacity is reached and further users attempting to call you will get a line busy message, the X.25 equivalent of the telephone's engaged signal.

Those members of the ODEX family that support native X.25 will, of course, allow you to define more than one X.25 line so, at a cost, network congestion need never happen.

X.25 and the OSI 7-layer reference model

X.25 defines layers 1, 2, and 3 in the OSI Reference Model. OSI is an ISO standard for worldwide communications that defines a networking framework for implementing protocols in seven layers. Control is passed from one layer to the next, starting at the application layer in one station,

proceeding to the bottom layer, over the channel to the next station and back up the hierarchy.

X.25 support in DI products

The following products provide native X.25 support:

ODEXplus

ODEX/400

ODEX/MVS

DINET

ISDN

Overview

ISDN (Integrated Services Digital Network) is a digital communications line that allows for the transmission of voice, data, video and graphics over standard communication lines.

The increasing need to transfer large volumes of EDI data, in particular CAD/CAM drawings, has focused attention upon high speed, low cost, communication. The traditional X.25 service over a Packet Switched Data Network (PSDN) has been a good general-purpose communications sub-system. However, its bandwidth and cost mean that it has become increasingly unsuitable for present day EDI requirements.

X.25 over ISDN provides both the transfer speed and cost benefits required by the new scenario in conjunction with the X.25 protocol. (X.25 is a communications protocol that defines both the structure of the data packets that comprise the protocol, as well as the manner in which they will be used.)

Two types of ISDN line interface are available, Basic Rate (BRI) and Primary Rate (PRI). An ISDN line consists of two different channel types:

- B-Channel – Carries bearer services, such as digital data, video, and voice.
- D-Channel – Carries control and signalling information and can also carry packet mode data.

BRI includes two 64 kbps (kilobits per second) B-channels and one 16 kbps D channel. PRI in Europe consists of 30 B channels and one 64 kbps D channel, while in North America and Japan 23 B channels and one 64 kbps D channel are provided.

ISDN Bearer services (those that permit data to be sent from one network to another, using only the lower three layers of the OSI model) can be offered in one of two transfer modes. Circuit mode offers a connection over a circuit switched network and provides a dedicated

end-to-end connection for delay-critical applications such as voice, video and real-time data. Packet mode offers a connection via a packet switched network.

Packet Switching

Packet Switched Network devices fall into two categories, Data Terminal Equipment (DTE), e.g. a PC, and Data Circuit-terminating Equipment (DCE), such as a modem or packet switch. These are normally located at the carrier's facilities.

The ability to offer packet switching services is broken down into two types: X31 Case A and X.31 Case B.

In scenario A, ISDN does not provide packet mode services but rather provides circuit mode access to a packet handler. In essence, the DTE at the user's site uses ISDN to establish a circuit switched connection to the Packet Switched Public Data Network's (PSPDN) DCE. Access to the trading partner's DTE is then conducted across the PSPDN using the ISDN B channel, as the D channel is reserved for signalling in this scenario (as it cannot carry the circuit switched data).

In scenario B, the PSPDN capability is 'embedded' into ISDN and packet services are offered. When a connection is formed, the packet handling function is this time part of the ISDN network. This allows the user (DTE) to see the local exchange as a DCE and in turn permits a connection between trading partners (DTE to DTE). As the local exchange has packet switch capability and will therefore be able to handle X.25 packets on arrival (i.e. the D channel does not have to be reserved for signalling), communication can take place on either the B or D channel. In the OFTP environment, the D channel is used.

ISDN support in DI products

The following products provide ISDN support:

ODEX Enterprise

DINET

BACS.*IP*

DEVILS

ODEXplus

ODEX/400

ODEX/MVS

CAPI

Overview

Common ISDN API (CAPI) is an application-programming interface used to access ISDN equipment. CAPI enables applications to access ISDN

adapters in a straightforward manner and allows unrestricted use of their functions through a standardized software interface. This interface, which offers a single point of access to different ISDN services such as data, voice and fax can then be used by applications.

The use of CAPI allows applications to communicate over ISDN lines without having to cater for differences in hardware manufacturers' specifications. Furthermore, in the event of future hardware developments, applications will remain unaffected, as CAPI will make any changes transparent to the application.

CAPI includes a number of important features including:

- Support for basic call features, such as call setup and call clearing
- Support for multiple B channels for data and/or voice connections
- Support for one or more BRI (Basic Rate Interfaces) and PRI (Primary Rate Interfaces) on one or more ISDN adapters.
- A transparent interface for protocols above Layer 3 in the OSI 7 Layer model.

The interface of CAPI is located between Layer 3 and Layer 4, and provides a point of reference for applications and higher-level protocols.

CAPI 2.0, which has replaced CAPI 1.1, allows applications to access multiple devices and multiple interfaces within those devices. Data Interchange Plc products supporting CAPI all support CAPI 2.0.

CAPI Messages

The communication between an application and CAPI occurs asynchronously using messages defined within CAPI. A message sent in either direction is always followed by an appropriate reply.

Messages initiated by the application are called Requests. A Request is followed by a corresponding message from CAPI called a Confirmation.

Messages initiated by CAPI are called Indications and the corresponding message from the application is called a Response.

The messages communicated are always ended by the appropriate suffix for that message type (`_REQ`, `_CONF`, `_IND` and `_RESP`).

Every message sent between the application and CAPI contains an identifying message number. When a Request is sent from an application to CAPI, the Confirmation sent in reply will always contain the same message number as the request. Indications sent from CAPI to the application are numbered uniquely and the application is prohibited from sending a Response unless it is in reply to an Indication. CAPI will ignore any such messages that are sent from the application.

The messages sent between applications and CAPI are split into three categories.

1. Messages concerning the ISDN signalling protocol, which are used on the D-channel
2. Messages concerning logical connections (B or D-channel)
3. Administrative messages

Of these three categories the signalling messages and the logical connection messages provide a useful insight into understanding how a request for a connection is managed.

Example Signalling Messages

Message	Description
CONNECT_REQ	Initiates an outgoing physical connection
CONNECT_CONF	Local confirmation of the request
CONNECT_IND	Indicates an incoming physical connection
CONNECT_RESP	Response to the indication
CONNECT_ACTIVE_IND	Indicates the activation of a physical connection
CONNECT_ACTIVE_RESP	Response to the indication
DISCONNECT_REQ	Initiates clearing down of a physical connection
DISCONNECT_CONF	Local confirmation of the request
DISCONNECT_IND	Indicates the clearing of a physical connection
DISCONNECT_RESP	Response to the indication

Example Logical Connection messages

Message	Description
CONNECT_B3_REQ	Initiates an outgoing logical connection
CONNECT_B3_CONF	Local confirmation of the request
CONNECT_B3_IND	Indicates an incoming logical connection
CONNECT_B3_RESP	Response to the indication

Message Queues

The actual communication between the application and CAPI utilises message queues. There is a single message queue for CAPI and one message queue for each registered application.

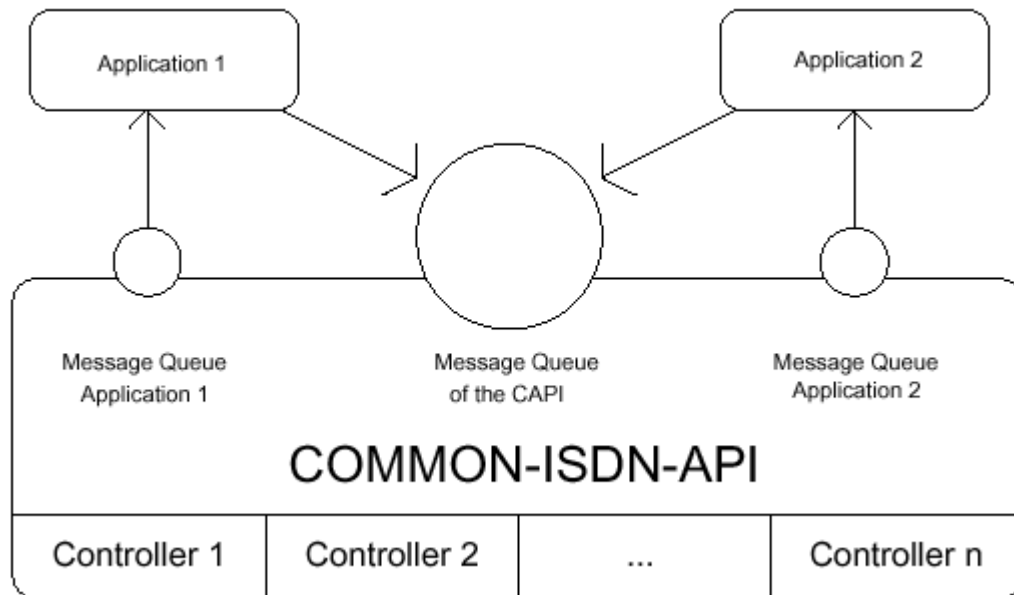
Messages sent from the application (Requests) are placed into the CAPI message queue for processing. In the reverse direction messages sent from an ISDN controller (Indications) are placed into the message queue of the addressed application.

Utilising Message Queues

Before an application can send messages to CAPI, it must first register itself with CAPI. This is performed using the CAPI_REGISTER function, which is called by CAPI to assign the application a unique ID and to setup the application's message queue.

All messages sent by the application are placed into the CAPI message queue, using the CAPI_PUT_MESSAGE function. In the event that the CAPI message queue cannot accept any more messages, this function will return an error.

CAPI manages the queue for each of the registered applications. Applications call the CAPI_GET_MESSAGE to obtain any new messages from this queue.

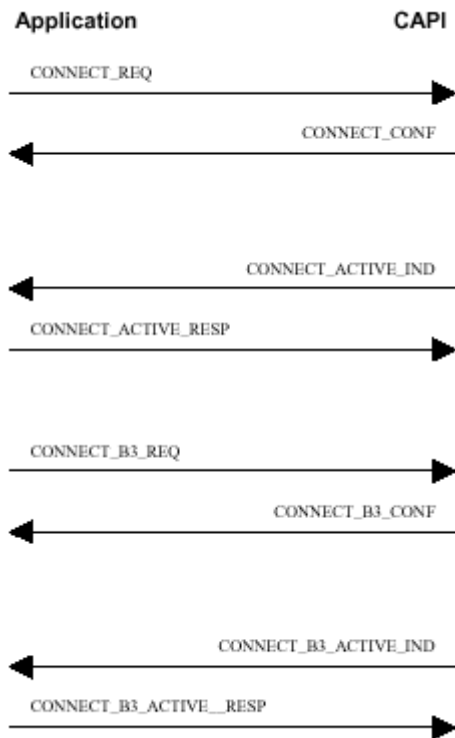


The CAPI functions such as the PUT and GET functions are made available to Windows applications in the CAPI2032.DLL. This DLL is copied to the local machine during the installation of the CAPI hardware and associated software.

CAPI Example: Establishing an outgoing call

The application first requests an outgoing physical connection and CAPI provides confirmation of this back to the application. CAPI then notifies the application that a physical connection has been activated and the application sends a response back.

The application then sends through a request to initiate a logical connection over the B-channel. This request is acknowledged by a corresponding confirmation from CAPI. CAPI then indicates to the application that a logical connection has been activated and the application sends a response back.



CAPI Configuration

As CAPI allows applications to control ISDN hardware devices, the configuration of a local CAPI device, such as an Eicon Diva Pro Card with ODEX Enterprise, is very straightforward. The installation of such a card will copy the required CAPI2032.DLL onto the machine and enable ODEX to control the ISDN interface.

RCAPI

The functions provided by CAPI can also be used to control remote hardware devices, such as a Cisco 801 or a Bintec X1000 router.

CAPI is currently only supported on the 800 series Cisco routers. The 801 has been successfully tested with ODEX applications and the 803 is also compatible.

The 801 router is not modular and so cannot accept additional hardware. It is limited to a single Ethernet connection and a single BRI connection (plus an RS232 console port).

The router needs to have IOS version 12.1(2)T or above

In some scenarios, for example when using a Cisco 801 router, additional third party software may be required to enable the application to remotely access the CAPI functions. When using the Cisco 801 router, the RVS-COM Lite software is required to allow a machine on the local area network to initiate calls on the Cisco router.

CAPI support in DI products

The following products provide CAPI support:

ODEX Enterprise

ODEXplus

DEVILS

TCP/IP

Overview

TCP/IP (Transmission Control Protocol/Internet Protocol) is the basic communication protocol of the Internet. It can also be used as a communications protocol in a private network (either an intranet, where parties within a single organisation can communicate, or an extranet, where parties may communicate with each other over an external but private network).

The two parts of the protocol, TCP and IP, were developed by the US Department of Defense (DOD) to solve the problem wherein a number of different networks, designed by different vendors, needed to connect with each other in what was, in essence, a network of networks. This network of networks became the Internet.

TCP/IP benefits

TCP/IP provides basic but high-demand services, such as file transfer and electronic mail (e-mail), across a very large number of client and server systems.

Several computers in a small company department can use TCP/IP (along with other protocols) on a single Local Area Network (LAN). The IP component provides routing from the department to the company network, then to regional networks, and finally to the global Internet.

TCP/IP is designed to be highly robust and automatically recover from any network node or phone line failure. In addition it employs flow control mechanisms, to allow for inadequacies of the receiving computer, and support for the detection of errors and lost data, with its ability to trigger retransmission until the correct data is correctly and completely received.

TCP/IP and the OSI 7-layer reference model

TCP/IP is a two-layer program, where TCP is the higher layer and IP the lower layer.

TCP sits in the Transport Layer of the OSI 7-layer model and provides all the packet and error-handling services of that layer.

IP sits in the Network Layer of the OSI 7-layer model and is responsible for the routing of data packets through the network. Each gateway computer on the network checks the packet address to see where to

forward the message. Just as in X.25, packets from the same message may be routed differently but will be reassembled at the destination.

IP addresses

Every machine on the internet, or within an intranet or extranet system, has a unique identifying number, known as an IP address. Internet IP addresses are globally unique and assigned by the Network Information Centre, whereas intranet and extranet IP numbers only need to be unique within their own network and can be assigned by an administrator.

The format of an IP address is a “dotted decimal number” such as:

216.27.61.137

The fact that internet IP addresses are globally unique allows IP networks anywhere in the world to communicate with each other.

The four numbers separated by decimal points are used to create classes of IP addresses that can be assigned to particular entities, such as businesses or governments, based on size and need.

Each IP address is divided into two sections: Net and Host. The first of the four numbers is always used to identify the network (Net) to which a computer belongs. The last of the four numbers is always used to identify the actual computer (Host) on the network. The second and third numbers may be used to identify the Net or the Host, depending on which class the IP address belongs to.

TCP/IP support in DI products

The following products provide TCP/IP support:

ODEX Enterprise

DARWIN 3

ODEXplus

ODEX/400

ODEX/MVS

DINET

XOT

Overview

XOT (X.25 over TCP/IP) enables X.25 packets to be sent over a TCP/IP network to an XOT-capable router, from where they are transported to the destination over an X.25 connection.

XOT support is provided on most ODEX platforms and allows users to connect to X.25 partners via an XOT router. In many cases, X.25 users prefer to install routers rather than have X.25 calls come directly into

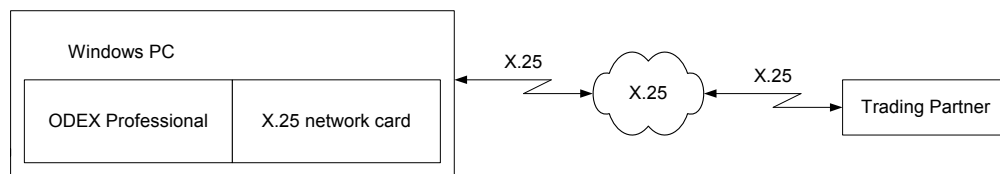
their comms applications, since the latter method can have security issues. Routers can be configured to block calls from unknown origins.

Where native X.25 is used, X.25 hardware, which can be costly, has to be connected to the machine on which ODEX is installed. However, where XOT is used, ODEX simply makes a TCP/IP connection to an XOT-capable router such as a Cisco, which can provide a sizeable cost saving.

Using an XOT router, ODEX can communicate with trading partners using either native X.25 or ISDN, depending on the routing rules and configuration of the router.

The example below shows how ODEX Professional, which had native X.25 support, would connect to an X.25 partner. Here the application would need X.25 hardware to be installed and configured. On Windows platforms, the cost of X.25 hardware may not be too great, but on mainframes and Unix machines it can be prohibitive.

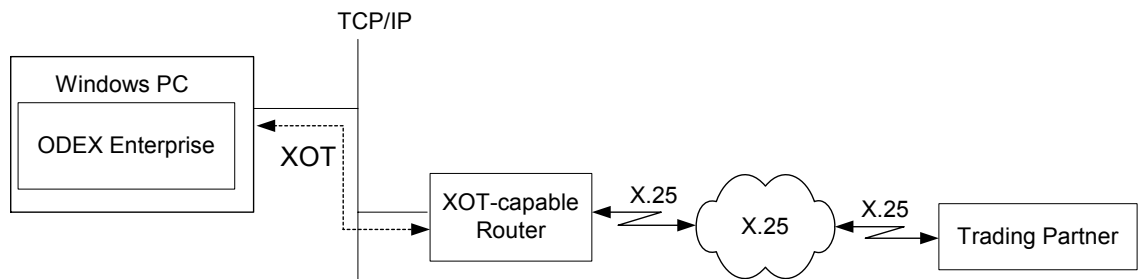
ODEX Professional and X.25



XOT scenarios

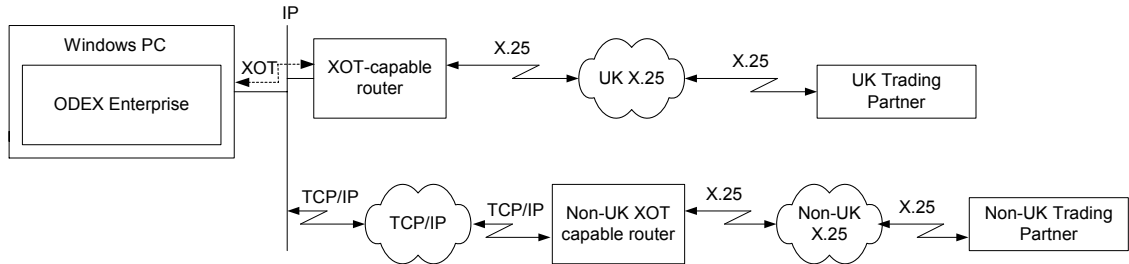
The diagrams below show a variety of XOT scenarios, using ODEX Enterprise as the example application.

ODEX Enterprise using XOT with an X.25 partner



In this scenario, ODEX Enterprise makes a TCP/IP connection to an XOT-capable device, such as a Cisco router. The router then makes an X.25 call to the destination, according to the information provided in the TCP/IP call.

ODEX Enterprise using XOT with UK and non-UK partners

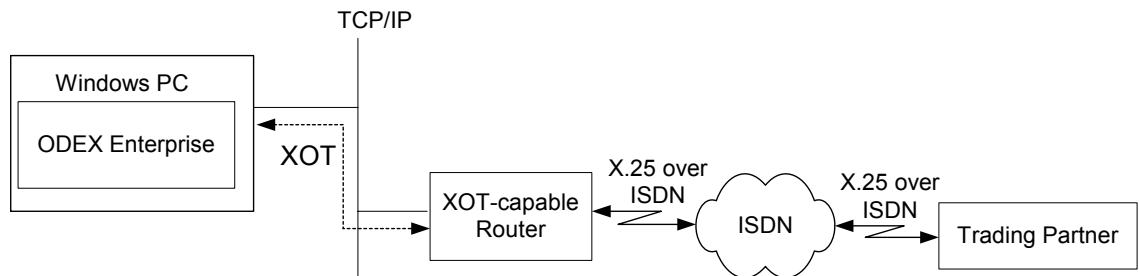


This example illustrates a possible scenario of a company making calls to UK X.25 partners using a router on their own local network, and making calls to non-UK partners via an XOT router on their partner's local network.

When connecting to non-UK partners, users can take advantage of a low-cost TCP/IP connection until the call reaches their partner's network. At this point the XOT router channels the data into the partner's X.25 lines, which provide the X.25 connection at local rates in that country.

In practice, this solution is likely to be used only by the largest corporations, which can install a router in each foreign location of their operations.

ODEX Enterprise using XOT with an ISDN partner



An XOT-capable router may also be able to handle ISDN calls. In this case, if the router is configured correctly, it can be used to route calls to both X.25 and ISDN partners. However, this is less suitable when a user has many ISDN partners, as each ISDN partner must be configured separately on the router. This is not the case for X.25 partners, as the information needed to make the connection to the partner (the X.25 NUA) is present in the data passed to the router from the ODEX application.

To achieve XOT with ISDN using a Cisco Router, the router requires a BRI (Basic Rate ISDN) interface. A Dialler interface is then configured on the Cisco, which essentially tells the router which number to dial. Further configuration is then required to get the Cisco router to use a particular Dialler interface (for each partner) when it receives a call from the X.25

NUA of the ODEX system. The Cisco then dials the number configured in the Dialler interface and connects over ISDN.

XOT support in DI products

The following products provide XOT support:

ODEX Enterprise

ODEXplus (using OFTP)

ODEX/MVS

PAD

A PAD, or Packet Assembler/Disassembler, is a device which enables remote users to gain access to the X.25 packet switching systems PSE (Packet Switch Exchange) using the telephone network.

All PTTs or X.25 service providers throughout the world have public PADs available for subscribers to gain access to their X.25 service.

X.28 communications require the use of a PAD. The PAD converts the X.28 asynchronous protocol to X.25 and acts as a buffer between the two systems, converting X.28 data to packets and vice versa.

X.28

X.28 is a communications standard providing asynchronous connections to an X.25 network. It requires a modem connection and an X.28 dial-up subscription to the local PTT or preferred VAN.

Asynchronous transmission means that information is sent in a continuous stream of characters as opposed to the more reliable synchronous transmission used by the "native X.25" standard. X.28 is cheaper than native X.25 but rather more error-prone.

At the start of a communication session, the user's software application will instruct the modem to make an outgoing telephone call to a PAD (Packet Assembler/Disassembler).

Once a connection to the PAD has been made, the PAD can be instructed to make an X.25 call to the trading partner of your choice. This is achieved in the same way as a native X.25 call would be made, by giving the destination's NUA (Network User Address, the X.25 equivalent of a 'phone number). Once the X.25 connection has been established then a two-way flow of data can be started.

The PAD acts as an interface between the asynchronous line, where data is transferred in a single stream, and the X.25 system, where data must be passed as packets. The PAD, as its full name suggests, takes data from the asynchronous phone line and assembles data packets to be sent and also takes packets from the X.25 service and disassembles them into a single asynchronous stream of data.

Unfortunately, PAD access is error prone as X.25 error correction standards do not cover this type of access because the X.25 connection terminates at the PAD (for X.25 over a dialup connection see X.32). Error correction on the line between the user and the PAD is the responsibility of applications at either end of the connection. (In the case of ODEX this is achieved by the use of OFTP Special Logic Extensions, which provides checksum functionality to check the integrity of received data and provides a mechanism for error recovery and data retransmission).

X.28 support in DI products

Owing to the relatively unreliable nature of X.28, DIP products no longer offer this method of communication.

X.29

X.29 is the CCITT procedure for the remote exchange of control information and user data between an X.25 host and a PAD. When an X.28 asynchronous user dials in to a PAD and establishes communication with a native X.25 host, the host may send X.3 parameters back to the PAD using the X.29 protocol. This lets the host system ensure that communications will be controlled by the correct forwarding characters etc., avoiding possible communications problems.

X.3

The X.3 standard deals with the first 18 parameters required for successful PAD operation.

Public PADs are used for many different functions, not just OFTP communications. They may, for example, be used as a logon to a mainframe service, which would have totally different communication attributes to the transfer of data. On an interactive terminal connection you would naturally wish to see what you were typing, and so the PAD would be required to echo all received characters back to you. This would be totally inappropriate for a file transfer system as it would double the data traffic and the cost of the connection.

For this reason a series of user-configurable settings known as the X.3 parameters (X.3 being the international standard), is set within the PAD to ensure that these communications attributes will be correct. Incorrect setting of these parameters will almost certainly result in communications problems as the two sides of an automatic conversation will not understand one another.

The X.3 standard defines the first 18 of these parameters but on some PADs there may be many more.

X.31

X.31 is an international CCITT specification for the connection of ISDN systems to X.25 networks. X.31 exists in two forms, X31 Case A and

X.31 Case B, both of which are described more fully in the Packet Switching section of the ISDN chapter. Here we describe Case B, which is supported by ODEX on most platforms. Case A is not supported by ODEX.

X.31 Case B is an international CCITT specification for the connection of ISDN systems to X.25 networks using the ISDN D-channel. This facility is available in Europe and Japan, but not in the UK.

The 16 kbps D-channel of an ISDN Basic Rate Interface (BRI) is primarily used for signalling between the ISDN and the router, but has some spare capacity which can be used for data communications. X.31 Case B services are offered by some ISDN providers, and give access to an X.25 network through the ISDN D-channel.

X.31 Case B services are primarily for organisations who want to connect to the public X.25 network at relatively low cost.

Transmission rates of a minimum of 9.6 kbps are offered over X.31 link, and the costs of operation are low when compared to the operation via an ISDN B-channel. This makes X.31 Case B a cost-effective solution for low rate data applications such as e-mail or credit card validations.

X. 31 support in DI products

The following products provide X.31 support:

ODEX Enterprise

ODEXplus

ODEX/MVS (though requires protocol converter hardware. ODEX/MVS understands only X.25 so ISDN lines have to be connected to the mainframe via boxes that map X.25 to X.31).

X.32

X.32 is a CCITT specification for a form of X.25 that is designed to operate over a dial-up (rather than leased line) connection to a public data network. It is specifically concerned with options to identify the user to the network.

In effect, X.32 is a special form of X.25, in that your computer may dial the X.25 network as and when you wish to use X.25. Unlike X.28, an X.32 connection allows synchronous X.25 to run over the dialled line. The advantages of this are that, whilst you are connected, you may both make and receive calls and can have multiple simultaneous sessions with the outside world over a single connection.

The X.32 recommendations are specifically concerned with options on identifying the user to the network. As the connection is dialled, theoretically anyone could get into the system and pretend to be this user, with all the attendant security problems. X.32 specifies a set of

login criteria and passwords that ensure the connected user has the authority to become an extension of the X.25 network.

X.25 HDLC (Higher Data Link Control) protocol standards protect the integrity of data over the vulnerable telephone link that connects your computer to the PSE. This means that any error correction is made at a lower level and is no longer the responsibility of the application software. Errors will be found and corrected faster and more efficiently.

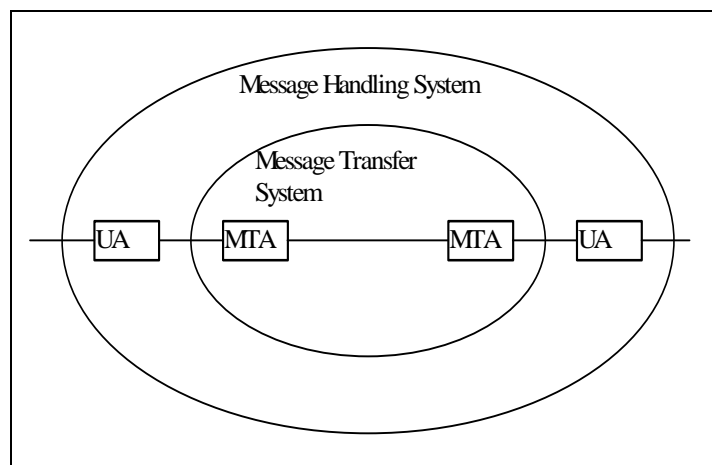
Typical line speeds for this type of connection range from 2400bps to 9600bps. X.32 is still not available in many countries and is not currently supported by DI products.

X.400

The X.400 protocol is not a single protocol but a collection of X.4nn protocols intended to handle documents of any type from simple mail messages right up to full scale EDI data transfers.

This protocol wraps the data to be transferred in an electronic “envelope” that has a function similar to the paper envelope used in the postal system. The envelope keeps the information together, keeps it safe from loss or corruption, and holds the address of the final destination to which the contents must be conveyed.

The X.400 protocol has two basic areas of definition, the message handling system and the message transfer system. The message handling system defines the way that messages are presented to the “user”, known as the UA or User Agent. The message transfer system defines the protocol used to transfer messages across the communications link between two Message Transfer Agents or MTAs.



There are two main approaches to EDI using X.400. The P1/P2 approach is the simpler where P1 is the name for the envelope and P2 is the Inter-Personal Message or content of the EDI file. The more complex

X.435 (sometimes called PEDI) approach uses the X.400 concept of EDIMs (EDI Messages) and EDINs (EDI Notifications) to control the flow of information.

X.400 support in DI products

The following products provide X.400 support:

ODEXplus (on SCO and AIX platforms only)