

T A N D E M
SYSTEMS REVIEW

VOLUME 8, NUMBER 1

APRIL 1989

BRANDIFINO



Terminal Connection Alternatives

NetBatch

NonStop CLX

5200 Optical Storage Facility

The OSI Model

Volume 5, Number 1, April 1989

Editor
Susan Wayne Thompson

Associate Editor
Anne Lewis

Special Projects
Ellen Marielle-Tréhöuart

Technical Advisors
Mark Anderton
Bart Grantham

Assistant Editor
Sarah Rood

Art Director
Janet Stevenson

Cover Art
Cara Koenig

Production and Layout
Claire Dolan
Niklas Hallin
Laurie Menden

Typesetting
Annie F. Valva

The *Tandem Systems Review* is published by Tandem Computers Incorporated.

Purpose: The *Tandem Systems Review* publishes technical information about Tandem software releases and products. Its purpose is to help programmer-analysts who use our computer systems to plan for, install, use, and tune Tandem products.

Subscription additions and changes: Subscriptions are free. To add names or make corrections to the distribution database, requests within the U.S. should be sent to Tandem Computers Incorporated, *Tandem Systems Review*, 18922 Forge Drive, LOC 216-05, Cupertino, CA 95014. *Requests outside the U.S. should be sent to the local Tandem sales office.*

Comments: The editors welcome suggestions for content and format. Please send them to the *Tandem Systems Review*, 18922 Forge Drive, LOC 216-05, Cupertino, CA 95014.

Tandem Computers Incorporated makes no representation or warranty that the information contained in this publication is applicable to systems configured differently than those systems on which the information has been developed and tested. It also assumes no responsibility for errors or omissions that may occur in this publication.

Copyright © 1989 Tandem Computers Incorporated. All rights reserved.

No part of this document may be reproduced in any form, including photocopy or translation to another language, without the prior written consent of Tandem Computers Incorporated.

CLX, ENFORM, ENVOY, EXT, GUARDIAN, MULTILAN, NetBatch, NonStop, SNAX, TACL, Tandem, the Tandem logo, TXP, and VLX are trademarks and service marks of Tandem Computers Incorporated, protected through use and/or registration in the United States and many foreign countries. The use of the symbol ® indicates registration in the United States only; registrations may not have issued yet in other countries.

IBM is a registered trademark of International Business Machines Corporation.

1 Preface

2 Terminal Connection Alternatives for Tandem Systems

Justin Simonds

20 NonStop CLX: Optimized for On-line Transaction Processing

Dan Lenoski

28 NetBatch: Managing Batch Processing on Tandem Systems

Dean K. K. Wakashige

38 Tandem 5200 Optical Storage Facility: A Hardware Perspective

Anand Patel

46 Tandem 5200 Optical Storage Facility: Performance and Optimization Considerations

Steve Coleman

54 Technical Paper: The OSI Model: Overview, Status, and Current Issues

Andrew Dunn

73 Index

Correction:

Please note the following correction to Figure 2 of the article titled "SCP and SCF: A General Purpose Implementation of the Subsystem Programmatic Interface," which appeared in the October 1988 issue.

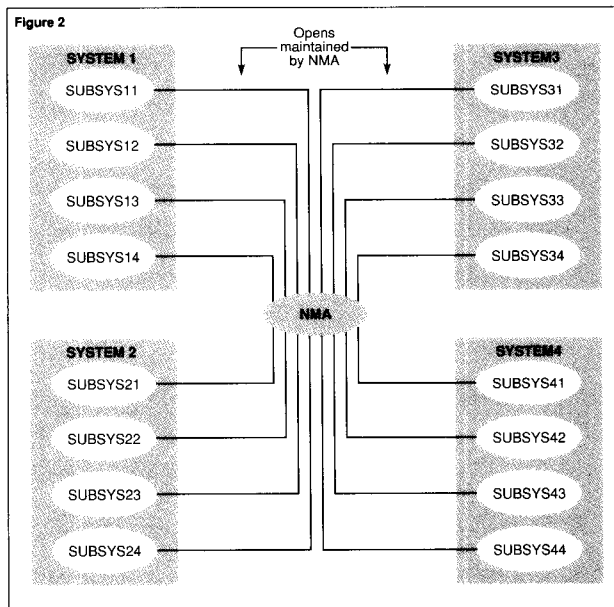


Figure 2.
Process-to-process opens without SCP.

The computer industry has made significant progress in developing both advanced network topologies and interconnected networks. The traditional star networks are rapidly evolving into distributed groups of subnetworks that are built with different networking technologies and supplied by different vendors. Although this trend toward interconnected networks and specialized workstations facilitates business solutions, connecting products from a number of vendors becomes a critical issue. Intersystem connectivity is the theme for a series of articles in the 1989 issues of the *Tandem Systems Review*.

The increasing number of specialized workstations makes the ability to easily connect different types of terminals especially important to multipurpose systems. The paper by Simonds compares the terminal connection products offered by Tandem. It reviews the hardware and software access methods and the characteristics of each alternative. The author also discusses the most important terminal connection requirements to consider when selecting an access method.

In geographically distributed processing, midrange systems for the office environment are assuming an increasingly important role. They offer economy and convenience by managing local microprocessor-based interfaces and, at the same time, assume the role of a distributed node in a larger network. The paper by Lenoski describes the hardware design considerations for the NonStop CLX system. The author also discusses the system's software, performance, and maintenance.

Batch processing continues to be a critical part of day-to-day operations in on-line transaction processing (OLTP) applications. When integrating batch applications with OLTP, issues such as automatic program execution and controlled use of system resources become important to economical system management. Wakashige's paper gives an overview of the NetBatch product. The author discusses the program's major components, functions, and interfaces.

The next two papers by Coleman and Patel follow Sabaroff's *Tandem Systems Review* paper (February, 1988) on the role of optical storage in information processing. The paper by Patel is a description of the 5200 Optical Storage Facility's hardware components, with a discussion of how the performance of the individual components facilitates the storage and access of archived information. Coleman's paper provides a performance analysis of the 5200 that can be used to size and predict the performance of an optical disk-based application.

The technical paper by Dunn gives an overview of the Open Systems Interconnection (OSI) reference model and highlights the most significant OSI issues. The OSI model was designed by international committees to provide interworking between heterogeneous computer systems. A complete set of protocols is now in place for several application areas, and the reference model is undergoing continual enhancement to incorporate new technologies and requirements. The OSI model enables users to install the equipment that is most suitable for the task, regardless of vendor or country of manufacture.

Susan Wayne Thompson
Editor

Terminal Connection Alternatives for Tandem Systems

Tandem allows a wide range of terminals and terminal configurations to be connected to its systems. This flexibility means that the customer has a number of communications controllers and processes to choose from when selecting a terminal configuration.

This article discusses terminal connection products offered by Tandem and compares their features. It lists the hardware and software methods available for connecting terminals to the Tandem system and the characteristics of each alternative. This information is of interest to the system managers, designers, integrators, and analysts who are concerned about choosing the most efficient and most economical method of attaching terminals to Tandem systems.

The article addresses the following topics:

- Preliminary information gathering.
- Tandem communications controllers and their characteristics.
- Tandem software options, including a table relating the input/output (I/O) processes to the various controllers.
- An overview of ways to connect nonstandard or special-application terminals.
- Examples of alternatives in connecting a number of 6530 (or compatible) terminals.

Information Gathering

The first step in selecting a connection method is to find out as much as possible about the proposed terminals and installation. This includes information about the type, number, and location of the terminals; the type of application running on the terminals; performance and reliability requirements; and cost constraints. Also, company guidelines must be examined before a terminal connection method can be selected.

Once the main areas have been identified, the acceptable range for each variable must be established, and the items ranked by importance. Figure 1 is an example of a worksheet that can be used for this purpose.

Type of Terminals

The type of terminal to be connected to the system must be considered as well as the short-term and long-term terminal acquisition plans. Terminal changes and upgrades require more flexible protocol and controller combinations.

Number of Terminals

The number of terminals influences the type of connection selected. Issues such as growth plans and the cost per terminal connection should be considered when selecting a connection method. For example, some connection methods geared toward a large number of devices are not economical when just a few devices are to be connected.

Location of Terminals

It is important to know both the distribution and locations within a building and the distribution and locations throughout the city, country, or world. The location of the devices is important when evaluating which connection methods are available. The cabling or telephone equipment needed can affect the cost significantly, and at times, the cost of cabling and telephone circuits alone actually determine the connection method.

Type of Application

The type of application that will be running on the terminals influence the choice of a connection method. For example, when developing production applications, development terminals usually run TACL™ (Tandem Advanced Command Language), Tandem subsystems such as the EDIT utility, and compilers, while production terminals usually run user-written, block-mode applications. Defining whether the terminals are used conversationally (TACL and line-at-a-time applications) or in block or page mode (PATHWAY transaction processing system) can affect the software module chosen to connect these terminals.

Performance Requirements

There may be some performance requirements that affect the choice. Generally, response requirements are established as 1 or 2 seconds. The user should determine what is actually required because fixed response requirements generally cost more. For an ATM or airline reservation system, a range of 2 to 7 seconds may be considered reasonable. For electronic mail users, an occasional 10- to 20-second wait may be acceptable if thousands of dollars are saved. On the other hand, military, hospital, or nuclear applications may have rigid response-time requirements.

Figure 1

Number of terminals: _____
Type of terminals: _____

If mixed, provide the number and location of each type.
Terminal protocol: _____
Special requirements: _____

(such as graphics or XON/XOFF flow control)
Response time requirement: Range _____ or Fixed _____
Single or multiple locations: _____

If multiple, specify number and mix of terminals, distance apart.
Any location-dependent requirements: _____

Terminal applications: _____
Transaction rate and message sizes: _____
Terminal users:
Data entry _____ Programmer _____ Operator _____ Manager _____
Reliability requirements: _____

Reliability Requirements

The degree and type of fault tolerance must be established because reliability requirements influence the choice of a connection method. For example, during a component failure, the number and location of the devices that also fail can vary with the configuration.

Figure 1.
Terminal worksheet.

Figure 2

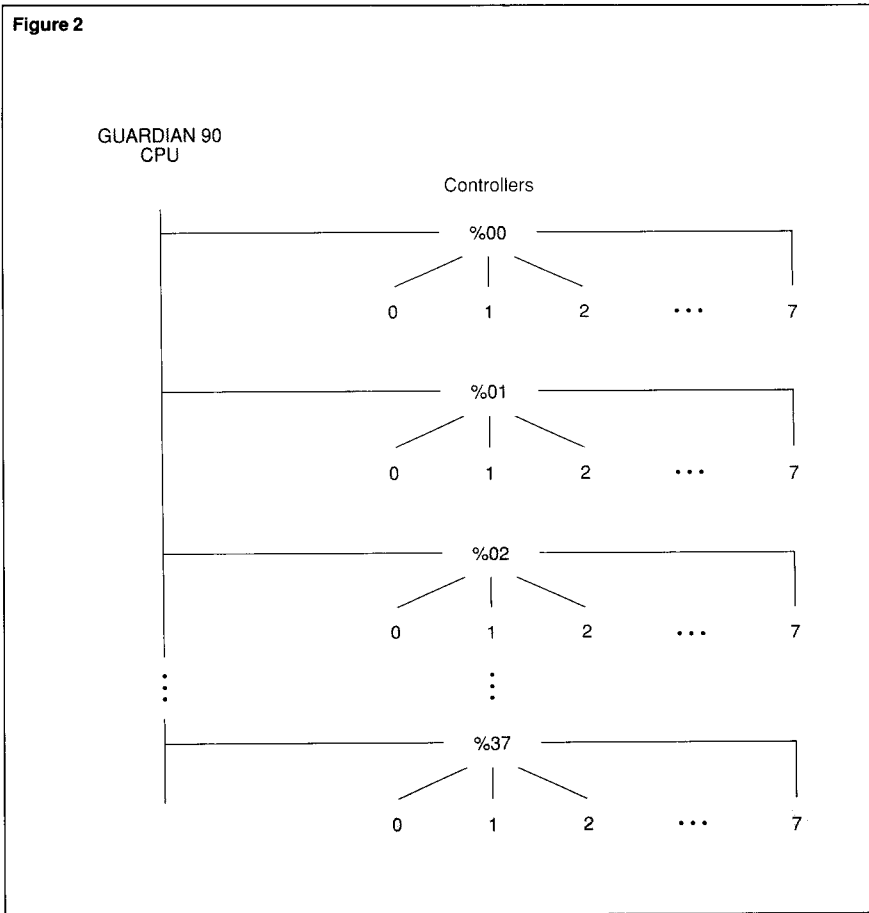


Figure 2.
Tandem I/O channel and
controllers.

Cost Constraints

Selection of a terminal connection is also determined by budget constraints and the priority the cost has in determining the method of connectivity. Cost constraints may well affect the selection. As in the previous description on reliability requirements, a choice can be made concerning the numbers of devices affected during a component failure and the associated costs of increased levels of fault tolerance. Ideally, only a single device would be affected during a component failure; however, the best business decision is one where the cost is weighed against the practical impact on business.

Company Guidelines

Finally, the company's guidelines and design criteria must be known. If two choices meet the requirements, it is important to make a decision based on company policy or management directives. For example, objectives such as a single vendor solution, state-of-the-art technology, or compliance with standards may be important considerations. This is also important when establishing or adhering to the targeted budget.

Tandem Communications Controllers

When connecting terminals to Tandem systems, Tandem's communications controllers offer a range of configuration possibilities. The following discussion describes several communications controllers and emphasizes their features and constraints.

The Tandem communications controllers and subsystems include:

- 6303/6304 asynchronous controller.
- 6202 byte-synchronous controller.
- 6204 bit-synchronous controller.
- 6100E communications subsystem.
- 6105 communications controller/6106 asynchronous communications controller.¹

Table 1 provides a summary of various Tandem communications controllers and their characteristics. Each controller and the Tandem I/O channel are described in detail in subsequent paragraphs.

Tandem I/O Channel

The current Tandem I/O channel allows 256 devices per channel, implemented as 32 controllers with eight ports each. Because not all Tandem controllers support eight devices, the controller mix further limits the actual number of devices that can be supported on a particular channel.

¹The smaller 3605 communications controller and 3606 asynchronous communications controller are the components designed for CLX systems. These components are virtually identical to the 6105 and 6106 communications controllers, except for size; everything discussed for the 6105 and 6106 controllers is applicable for the 3605 and 3606 controllers.

Devices are addressed on the I/O channel by controller number and device number. (See Figure 2.) A disk device with an address of %10 translates to controller %1, device %0. The user defines these addresses in the configuration file that is used by the INSTALL utility during the SYSGEN phase.² Addresses are provided to the customer by the field engineer. (See Figure 3.) These numbers are reflected by the peripheral utility program (PUP) when displaying the list device function (LISTDEV).

In the case of communication ports, the user may sometimes notice that two devices are required for a single physical port. For example, the user may be adding a 6202 byte-synchronous board and be given the address %100 by field service. This takes addresses %100 to %107. Although the board has four physical ports, there are eight "logical" ports because a read and a write port exist for each physical port. Dual ports on some communications controllers allow them to communicate in a full-duplex mode.³ (See Figure 4.)

6303/6304 Controller

The 6303/6304 asynchronous controller supports 32 half-duplex ports and both current-loop and RS232 interfaces. Each port has a maximum transfer rate of 19.2 Kbps. The controller supports a variety of terminals; for example, all of the Tandem terminals (such as the 6530, 6520, and 6510) and other vendors' terminals (such as the Zentec, ADM-2, Beehive, and HP). Generally, any type of asynchronous ASCII device can be attached.

Individually Configurable Lines. Each terminal installed with the SYSGEN program on a 6303/6304 controller is given a single address. Therefore, the port is half-duplex; either a read or a write can be done against the device, but both operations cannot be done at the same time. This is the only Tandem communications controller that is inherently half-duplex. Full-duplex can be supported on this controller, but this requires two physical ports (one for read operations and one for write operations) and a special cable to link the two ports.

Table 1.

Controller summary. Note that the maximum number of supported terminals is a function of the I/O process; the limits quoted are logical but not necessarily practical.

Controller	Number of ports	I/O addresses	Number of I/O slots	Maximum terminals supported	Diagnostics
6303/6304	32	32	3	32	SHADOW
6202	4	8	1	1020 (255x4)	SHADOW
6204	4	8	1	1020 (255x4)	SHADOW (TMDS)
6100E					
6120-series LIU	15	64 (double-ported CIU)	2	15 async, 3825 sync (255x15)	DIAG6100
6120-series LIU	15	32 (single-ported CIU)	2	15 async, 3825 sync (255x15)	DIAG6100
6140-series LIU	60	64 (double-ported CIU)	2	60	DIAG6100
6140-series LIU	60	32 (single-ported CIU)	2	60	DIAG6100
6105	4	8	1	4	TMDS
6106	16	8	1	16	TMDS

Figure 3

```

CONTROLLERS:
  ASYNCA 6303 0, 1 %040;
  BYTEA 6202 0, 1 %100;
  BITA 6204 0, 1 %300;
  ICC01 6105 2, 3 %300; (requires PATHS paragraph)

PATHS:
  ICC01P ICC01;

PERIPHERALS:
  $A00 ASYNCA.00 TERM ^^ 6530; (%40
  $A01 ASYNCA.01 TERM ^^ 6530; (%41 One address
  . . . . . per port
  . . . . .
  $A31 ASYNCA.31 TERM ^^ 6530; %77)
  $LHLA BYTEA.0,BYTEA.1 AM6520 ^^ LINE; (%100 & %101 Two addresses
  $LHPHX BYTEA.2,BYTEA.3 AM6520 ^^ LINE; %102 & %103 per port
  $LHLAT BYTEA.4,BYTEA.5 X25 ^^ LINE; %104 & %105
  $LHLB BYTEA.6,BYTEA.7 X25 ^^ LINE; %106 & %107
  $$SNAX1 BITA.0,BITA.1 X25 ^^ LINE; %300 & %301
  $$SNAX2 BITA.2,BITA.3 X25 ^^ LINE; %302 & %303
  $$SNAX3 BITA.4,BITA.5 SNAX ^^ LINE; %304 & %305
  $$SNAX4 BITA.6,BITA.7 SNAX ^^ LINE; %306 & %307
  $ATP00 ICC01P.0 ATP ^^ 6530; %300 & %301
  $AM650 ICC01P.1 AM6520 ^^ LINE; %302 & %303
  $X2500 ICC01P.2 X25 ^^ LINE; %304 & %305
  $$SNAX5 ICC01P.3 SNAX ^^ LINE; %306 & %307

```

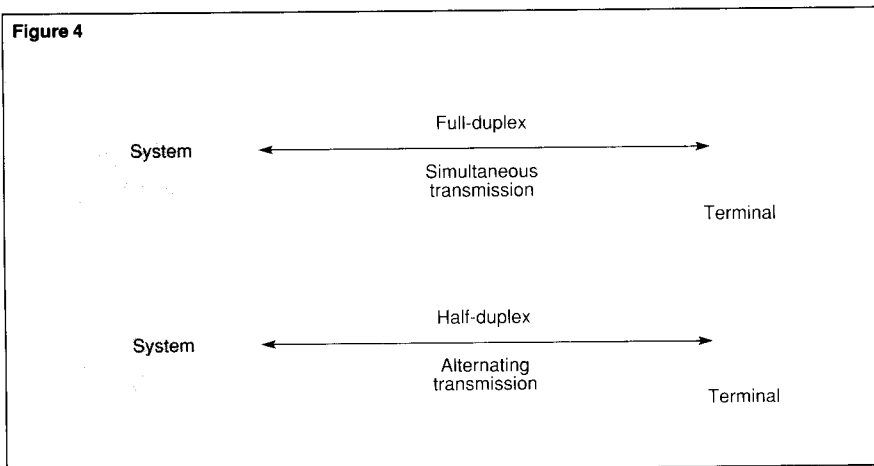
Figure 3.

Tandem INSTALL configuration file example. Note that because this is an example, details

necessary for actual system generation have been excluded.

²The INSTALL/SYSGEN program automatically installs new system software and includes user prompts to allow for easy customization of configuration and system generation. INSTALL/SYSGEN eliminates costly errors by ensuring consistency.

³Full-duplex is defined as the ability to transmit data simultaneously in both directions. Half-duplex is defined as the ability to transmit in one direction at a time.



```

CONTROLLERS:
  ASYNCA 6303 0, 1 %040;

PERIPHERALS:
  $A00 ASYNCA.00 TERM ^^ 6530; ! define for asynch 6530 port
  $A01 ASYNCA.01 TERM ^^ 6530;
  $A02 ASYNCA.02 TERM ^^ 6530;
  . . .
  $A30 ASYNCA.30 TERM ^^ 6530;
  $A31 ASYNCA.31 TERM ^^ 6530;

```

Figure 4.
Duplex definitions.

Figure 5.
6303/6304 INSTALL configuration file example.

Terminals connected to a 6303/6304 controller do not use qualified GUARDIAN® 90 names since only one terminal is allowed on a port. A qualified name is the extended name that is used when it is necessary to identify devices on the system. For example, the fully qualified name of a disk file is:

`\<system name>.$<volume name>.<subvolume name>.<file name>`

In the case of some other controllers, a *sub-unit* (similar to a subvolume name) is required because other controllers allow multiple devices to share the same port. A device cannot be uniquely identified by the port (or line) name alone and must use the additional qualification of a subunit name (e.g., `$<line name>.#<device name>`). In most instances where this is the case, some configuration beyond the SYSGEN procedure is required. Normally this is accomplished through a utility subsystem such as Tandem's CMI (Communication Management Interface) or SCF (Subsystem Control Facility).

As shown in Figure 3, terminals installed with the SYSGEN program under the 6303 controller ASYNCA are known by that name (e.g., \$A00; no subdevice qualifier is needed or even exists). This means that the configuration of these terminals can be handled by SYSGEN; after the system is loaded, terminals are available without the user having to configure anything additional. The terminals can be addressed without the use of any qualification; for example, `#<subunit>` is not needed in the device name to identify the terminal.

The 6303 controller ASYNCA has two types of boards: the 6303 asynchronous controller and the 6304 asynchronous extension. The asynchronous controller contains all the board logic and I/O control along with two asynchronous ports, and it can support up to two extension boards. The extension boards are port-only logic supporting up to 15 ports; they are not individually recognized by the system, and as a result, the (potentially) three boards are installed with the SYSGEN program as a single board. (See Figure 5.)

The 6303 controller ASYNCA appears to the SYSGEN program as a single controller with 32 attached devices but to the hardware system as four separate controllers. In this way, terminal \$A00 in the example would have an address of %40, \$A07 is %47, and \$A08 is %50. Therefore, the addressing for a fully configured asynchronous controller takes 32 addresses or four logical controllers of 8 each. In the example, %40 through %77 are allocated addresses on that I/O channel. This holds true even if the addresses are not used.

Also, because the 6303 and 6304 boards must be adjacent in the I/O bay, only certain slots can be used for these boards. Even though there are three slots vacant in an I/O bay, they may be unusable for 6303/6304 controllers without another I/O cabinet.

Fault Tolerance. The 6303/6304 requires three physical slots for every 32 devices. This could require the purchase of an additional I/O cabinet, power supplies, and patch panel space. Thirty-two I/O channel addresses are used for every 6303/6304 controller. Reliability and maintenance could also be a consideration.

The controller is fault-tolerant to the board level. This means that the controller, the extension card, or the port are possible causes of a failure. If the controller fails, all 32 lines on that controller set fail; if an extension card fails, it could take down 15 ports; and if an individual port fails, field service must take all lines on that controller down while the board is being replaced.

Additionally, the 6303/6304, 6202, and 6204 controllers use SHADOW diagnostics, a field service utility that diagnoses problems on the Tandem system. Because the utility uses its own operating system instead of the Tandem GUARDIAN 90 operating system, a CPU must be disabled and then loaded with the SHADOW utility to give the utility direct I/O channel access to the controller being diagnosed. This means that all devices on the controller must also be temporarily disabled.

Flow Control. An item also worthy of consideration is the type of flow control used by the 6303/6304 controller. Flow control is the ability to control the rate of data transmission. If one end of the transmission is having buffer problems, it needs to have the ability to temporarily stop the transmission from the other device. When the buffer problem is resolved, it also needs the ability to resume the transmission. There are several popular forms of flow control:

- Clear To Send (CTS) is done using the RS232 interface. When CTS is up, or high, the other side may transmit; when it is dropped, or low, the other side stops transmission.
- Data Terminal Ready (DTR), another RS232 signal, is similar to CTS. If the device can receive data, DTR is up. If the device is unavailable, DTR is low.
- XON/XOFF flow control method uses a special character, known as an XOFF, to temporarily suspend transmission. When transmission can resume, an XON character is transmitted.

A fourth method of flow control, called the TPAUSE and Reverse TPAUSE flow control, is used by the 6303/6304 asynchronous controller. TPAUSE is a method of flow control unique to Tandem. TPAUSE also uses the RS232 interface, but uses pin 12, which is often unused by RS232. When pin 12 is low, transmission occurs; when pin 12 is high, transmission is suspended. Reverse TPAUSE is the same, except the pin states are reversed: a low state suspends transmission and a high state allows it. Due to the lack of full-duplex compatibility, the 6303/6304 asynchronous controller has no XON/XOFF support.

6202 Byte-Synchronous Controller

The 6202 byte-synchronous controller is a high-speed controller that can support data rates of 56 Kbps with a maximum aggregate rate of 160 Kbps (e.g., the combined rate of all four lines cannot exceed this limit). The controller supports four half-duplex or full-duplex RS232 or RS422 communication lines and takes eight I/O channel addresses. Like the 6303/6304 asynchronous controller, if a bad port is discovered, all lines must be disabled to change the controller.

The 6202 board supports a wide variety of byte-synchronous protocols, such as X.25 and Tandem's AM6520 protocol, which can be mixed in any combination on the board. The controller does not allow asynchronous or bit-synchronous protocols. If configuring a mix of asynchronous, byte-synchronous, and bit-synchronous protocols using the 6303/6304, 6202, and 6204 controllers, there may be available ports remaining on each board.

6204 Controller

The 6204 bit-synchronous controller is a high-speed controller that supports four half-duplex or full-duplex RS232 or RS449 communication ports. The ports can be configured to support data rates of 220 Kbps with a maximum aggregate rate of 500 Kbps. The combined rate of all four lines may not exceed this limit; the maximum speed on a single full-duplex line is 220 Kbps. The board supports a variety of bit-synchronous protocols, such as X.25, SNAX™/XF (SNA communication services—extended facility), and ENVOYACP/XF (ENVOY™ bit-oriented protocols with extended functions), and takes eight I/O channel addresses.

The 6204 controller is similar to the 6202 and the 6303/6304 controllers in its aspects of being fault-tolerant to the board level. This means that a controller failure disables all four communication ports, or the failure of a single port requires the disabling of the three remaining ports to correct the problem. The 6204 controller, like the other two controllers, is diagnosed by the SHADOW diagnostic utility.

The 6204 comes in two varieties: the 6204-1, which uses an RS232 interface, and the 6204-2, which supports RS449. The latter requires a special patch panel; one disadvantage is that if you have one line that needs RS232 and another that needs RS449, you need two controllers unless you use a black-box type of converter.

It should be noted that the 6303/6304, 6202, and 6204 controllers have virtually no buffer limitations. All buffering is done within the CPU itself.

6100E Communications Subsystem

The 6100E communications subsystem can support up to 15 LIUs (line interface units) in each cabinet bay. An LIU is made up of two boards: a CLIP (communication line interface processor) and an LIM (line interface module). A CLIP is responsible for the link level of a protocol (such as polling in the AM6520 environment, discussed later), while the LIM provides the physical interface (such as RS232, RS449, and V.35). (See Figure 6.) The LIUs come in two varieties, the 6120 series and the 6140 series. The 6120 series is a single-port LIU that can support any protocol. The 6140 series is a four-port LIU that only supports asynchronous protocols. The 6100E supports all standard Tandem protocols.

Fault Tolerance. The 6100E subsystem is fault-tolerant to the line level, which means that the only point of failure is the LIU. The 6100E has been designed for quick fault isolation. Diagnostics exist on-line (DIAG6100) on the 6100E, and customers are encouraged to take an active role in fault isolation and determination.

Fault tolerance is enhanced on the 6100E because it is made up of components known as FRUs (field replaceable units). If an operator runs diagnostics and determines that the problem is caused by a bad LIU (CLIP or LIM), the unit can quickly be replaced if there is a spare LIU on hand. If not, a less critical line could be taken down and its components (the LIU) used for a more critical down line. This is a unique capability that allows recovery in minutes rather than hours.

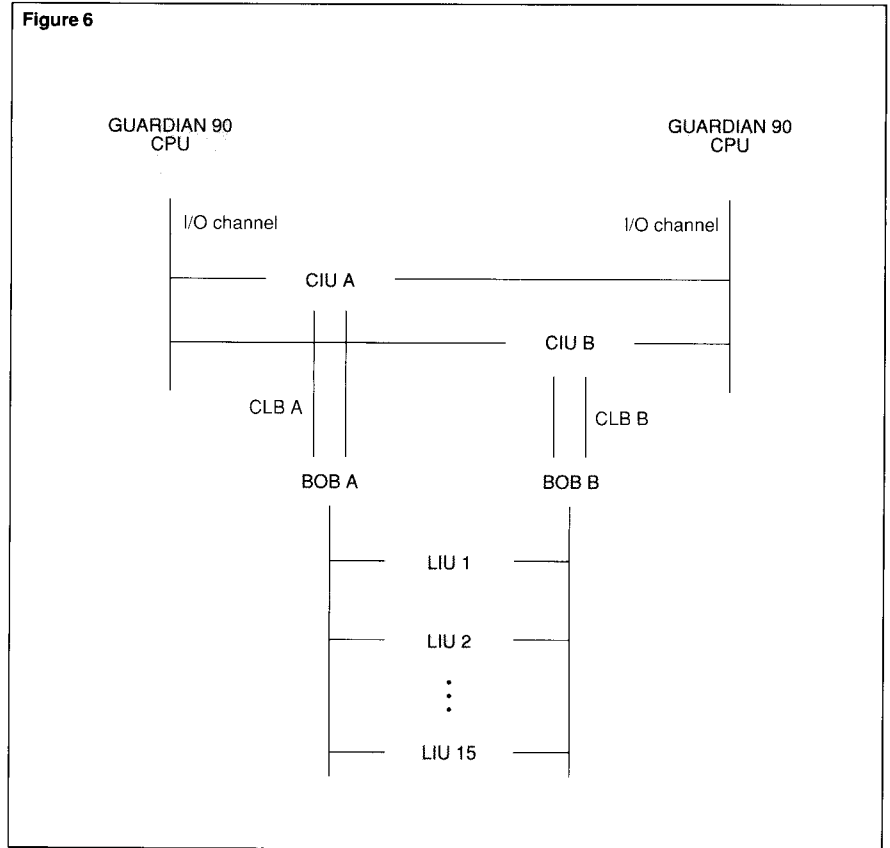
Flexibility. The 6100E is much more flexible because the LIU can support asynchronous, byte-synchronous, and bit-synchronous protocols (with the exception of the 6140-series LIU, which supports asynchronous only). A change of the access method does require a system generation, although this will no longer be necessary with the release of dynamic system configuration (DSC). (For details of the DSC product, refer to the *Dynamic System Configuration Reference Manual*.) If there are plans to convert to various protocols that will require a controller change, there could be a significant savings in the selection of a 6100E.

Performance. Another advantage is performance. On the 6100E, the physical and link layers of a protocol are handled by the LIU, which frees up CPU cycles. This is most notable in polling-type protocols where CPU utilization would otherwise be wasted on idle polling. A 6100E can sometimes pay for itself in the amount of CPU cycles it can give back to a system. For example, if the 6100E can reduce the CPU busy of a TXP™ processor by 25%, then 25% of the cost of the TXP processor can be subtracted against the cost of the 6100E. Because processors cost substantially more than communications subsystems, it can be a very cost-effective solution. Performance aspects must be examined in each case, as CPU utilization is protocol-dependent.

The 610X series of controllers has less CPU impact than the 6303/6304, 6202, and 6204 controllers, but they do have some buffer limitations. The 610X controllers have a 64-Kbyte memory that holds the link-level protocol. Any remaining memory is available for buffer space, which is affected by various parameters (such as FRAMESIZE and WINDOWSIZE) of the SYSGEN program. Due to the variance available within these parameters, a specific buffer size for each protocol is not available. Therefore, when selecting a controller, be aware of a potential buffer limitation within the 610X product line.

Configuration. The 6100E communications subsystem has cabling advantages that allow all the communication lines to connect to a single cabinet. The cabinet can be conveniently located near the telephone communications room or cable termination point, and maintenance activity can be carried out without disrupting other system cabling. The 6100E is ideal for large clusters of terminals located away from the system, as it can be located over 1600 feet from the Tandem system and does not require a computer room environment. Additionally, because the 6100E is attached using a fiber optic cable, it has very high security and is impervious to electrical interference.

By using the 6140-series LIU, a 6100E cabinet with three 6100s can connect up to 180 terminals while using only six system I/O slots. Although the 6100E only takes two I/O slots for the communications interface unit (CIU), it requires



64 I/O channel addresses. Where the 6303/6304 controller took addresses %40 to %77, the 6100E would take the addresses %40 to %137, twice the controller addresses of the 6303/6304, 6202, and 6204 controllers. (To support 15 ports, four 6202 or 6204 controllers would be needed, which requires 32 I/O channel addresses; however, four 6202 or 6204 controllers actually provide 16 rather than 15 ports.) All addresses are allocated even if the subsystem is not fully loaded, which means that for the standard configuration there is a maximum of three 6100 subsystems between two CPUs.

Figure 6.

The 6100 subsystem is made up of various components. A communications interface unit (CIU) is the I/O controller for the subsystem. A CIU-to-LIU bus (CLB) connects the controllers to the subsystem. A break-out board (BOB) multiplexes the data to and from the LIUs to the system.

Figure 7

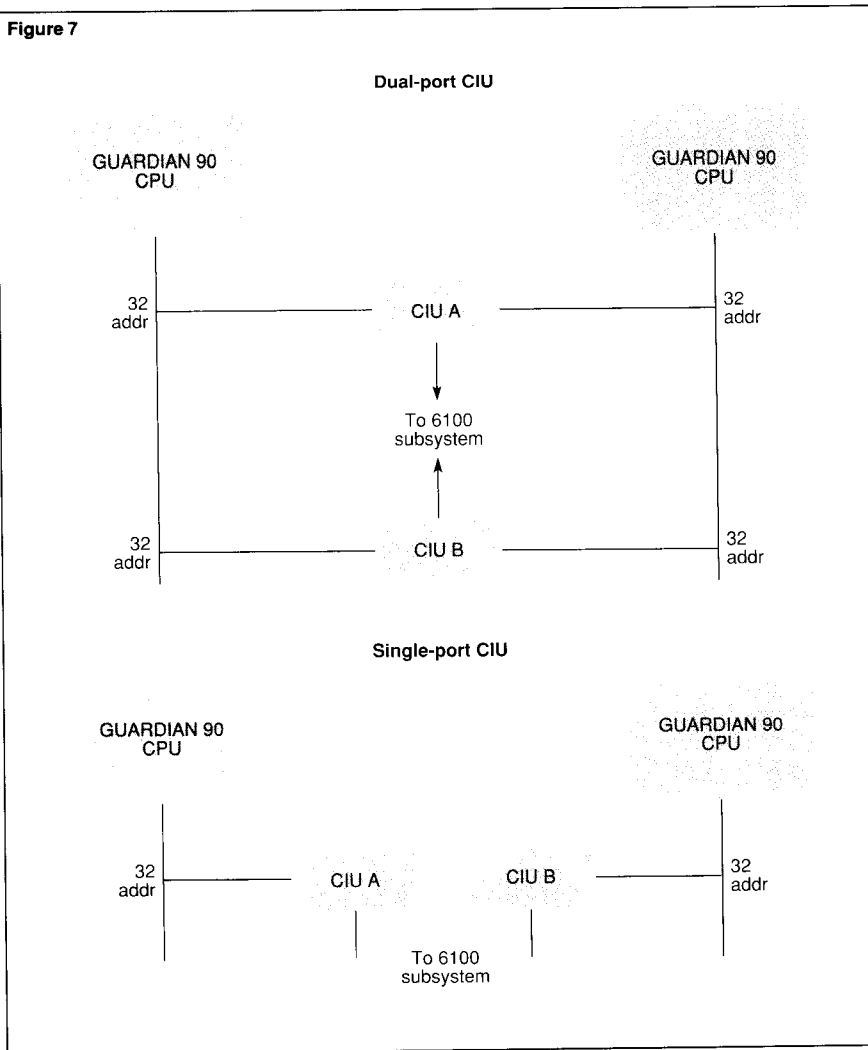


Figure 7.
Dual-port and single-port
CIU configuration.

The option SINGLE PORT CIU cuts the number of required addresses in half and sacrifices only a small degree of fault tolerance. Instead of connecting to two I/O channels, the CIUs connect to a single I/O channel. (See Figure 7.) Although this seems like a loss of fault tolerance, the 6100E still has access to both I/O channels, though not with each controller. As a result, a CPU and the CIU attached to the other CPU would have to fail to cause the failure of the 6100E. Using this optional configuration, the 6100E uses the same number of addresses as the older synchronous controllers and half the number required for the same number of ports using the 6303/6304, 6202, and 6204 controllers.

6105 and 6106 Controllers

Many of the benefits of the 6100E subsystem are incorporated into the 6105 communications controller (6105) and the 6106 asynchronous communications controller (6106). The 6105 and 6106 together offer the same performance benefits as the 6100E because the link-level protocol processing is done on the board. These two boards offer on-line maintenance with the Tandem Maintenance and Diagnostic System (TMDS). Additionally, special versions of the 6105 and 6106 controllers (the 3605 communications controller and the 3606 asynchronous controller) are available for the CLX system.

The 6105 and 6106 contain four programmable microprocessors (the CLIPs), each of which handles the protocol for one communication line. A patch panel provides the electrical interface to the communication lines. The boards use eight I/O channel addresses per controller. (See Figure 8.)

Because the 6105 and 6106 controllers are designed with a single board clock (a potential source of failure), they offer less fault tolerance than the 6100E subsystem. Any other failure results in a port (and its line) being disabled. However, changing the board requires that all four lines be disabled during maintenance. So again, as is required when fixing a bad port on the 6303/6304, 6202, and 6204, all lines on the board must be disabled. If this is not a problem, the 6105 and 6106 controllers may be the preferred choice.

6105 Communications Controller. The 6105 communications controller is a single-board controller capable of running any 6100E access method or protocol. It is similar to the 6202 and 6204 controllers, but it has the advantage of on-line diagnostics and protocol flexibility. The 6105 is equivalent to four 6120-series LIUs and supports four ports of any protocol.

6106 Communications Controller. The 6106 asynchronous communications controller takes one I/O slot and supports 16 asynchronous ports. The 6106 compares to four 6140-series LIUs. Addressing anomalies do not exist because the system addresses the 6106 as a single logical board. If given the address %40, only %40 through %47 is taken by this controller.

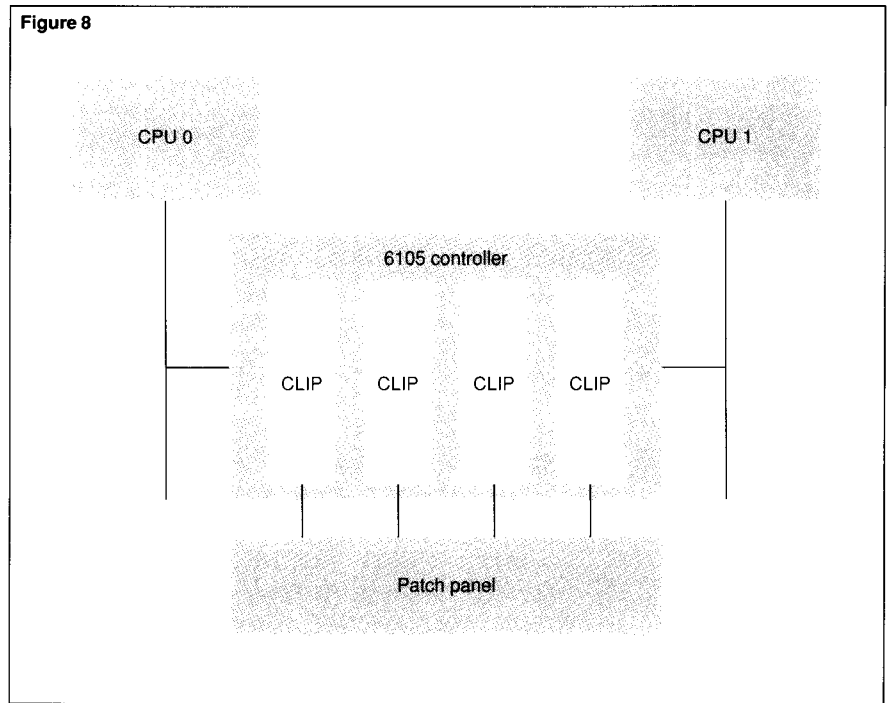
Full-duplex communication (data transmission in both directions simultaneously) is possible on the 6106, although Tandem does not have any full-duplex protocols currently released for the 6106. Flow control is accomplished by XON/XOFF, TPAUSE, or CTS methods. Terminals needing buffers greater than 2000 bytes require two ports of an LIU; in this case, only two terminals can be supported even though there are four ports. This is due to the extended buffer requirements of these terminals. Some speed combinations are not allowed on LIUs of the 6106. (Refer to the *System Generation Manual* for the limitations.)

Support offered by the 6106 is more efficient than that provided by the 6303/6304. The 6303/6304 requires 3 I/O slots and 32 I/O channel addresses to support 32 terminals, while the 6106 can support the same number of terminals by using only 2 nonadjacent I/O slots and 16 I/O channel addresses.

Tandem Communications Software

This section discusses the software portion of terminal connection to Tandem systems. Tandem offers a group of software products that support a wide range of standard and nonstandard terminals and their protocols. The product list supporting standard terminal connections includes:

- TERMPROCESS.
- ATP6100 (asynchronous terminal and printer processes for the 6100 communications subsystem).
- AM6520 (6520 device-specific access method).
- AM3270 (3270 device-specific access method).
- SNAX/XF.
- X25AM (X.25 access method).
- MULTILAN™.



The product list supporting nonstandard terminal connections include:

- ENVOYACP/XE
- SNAX High-Level Support (HLS).
- CP6100 (communication protocols for the 6100 communications subsystem).

Each terminal varies in the number and type of protocols that it can use for connection to the Tandem system. For example, an IBM SNA terminal requires SNAX/XF as its software interface, while a Tandem 6530 terminal can use TERMPROCESS, ATP6100, AM6520, SNAX/XF, or X25AM as a software interface.

Figure 8.

View of the 6105 controller. The combination of a CLIP and the patch panel equates to an LIU in the 6100E subsystem.

Table 2.

Summary of Tandem protocols and associated controllers. Note that for any of the protocols listed below running on a CLX system, a 36XX series controller must be used.

Tandem protocols	Associated controller
TERMPROCESS	6303/6304
ATP6100	610X series (6100E, 6105, 6106)
AM6520	6202, 6100E, 6105
AM3270	6202, 6100E, 6105
X25AM	6202, 6204, 6100E, 6105
SNAX	6204, 6100E, 6105
ENVOYACP/XF	6204, 6100E, 6105
CP6100	6100E, 6105

Table 2 is a summary of the various Tandem I/O processes and their relationships to Tandem controllers.

The following is a discussion of the different I/O processes. Note the details that may affect a decision, such as the number of devices that can be supported, the type of Tandem controller needed, and the protocol overhead.

TERMPROCESS

TERMPROCESS is a standard asynchronous I/O process capable of supporting 32 devices. Any kind of conversational (line-oriented) TTY terminal can be supported. Conversational-type applications include TACL and the Tandem utility subsystems such as EDIT, the file utility program (FUP), and PUP. There is no protocol involved other than an optional parity check of the characters.

TERMPROCESS also supports Tandem 6520 block-mode protocol, which is available for Tandem 652X and 653X terminals and 6530 PC emulation software packages. TERMPROCESS works with the 6303/6304 controller; its advantages and disadvantages were discussed previously. Each terminal is defined with a unique name and is fully configured by the SYSGEN program. (For a complete description of the block-mode protocol, refer to the *653X Multi-Page Terminal Programmer's Manual*.)

ATP6100

ATP6100 provides the same function as TERMPROCESS except that it works with the 6100E, 6105, or 6106 controllers. The most cost-effective solution is using the 6100E controller with the 6140-series LIU or the 6106 controller. These controllers allow the maximum number of terminals to be connected per controller.

Due to the way the 610X products have been developed, each terminal is known to the system by a qualified name:

\$line name.#subunit

where

line name is the port or LIU name.

subunit is a unique name within a particular LIU.

Even so, a terminal can be fully configured by the SYSGEN process and a terminal can be immediately active following a system load with no further configuration. ATP6100 also has a few additional features not found in TERMPROCESS, such as XON/XOFF support, that may be useful for some devices. Because the 610X controllers do more protocol work, there should be slightly less CPU impact using ATP6100.

AM6520

AM6520 is a byte-synchronous protocol, similar to the IBM BSC3270 protocol. It is used with Tandem terminals running in synchronous or asynchronous mode and allows for multiple terminals (up to 64) per port.

The terminals can be connected individually to a line, daisy-chained (the second terminal connected to the first, and so on), joined as separate drops on a multidrop line, or connected to a terminal cluster concentrator (TCC). Line sharing is possible because each terminal has its own address and is polled individually.

Polling is a technique used to control a line when multiple devices share the line. A line supervisor, in this case the Tandem processor or a 610X controller, sends a message (called a poll) to a device to see if it has data to transmit. If it does, that data is sent in response to the poll; if not, a "no data" message is returned and the line supervisor polls the next device.

AM6520 uses qualified names to relate an individual terminal to a specific polled address. CMI must be used to fully configure AM6520 devices after a cold load of the system. Therefore, terminal configuration is a two-step process.

The AM6520 process works with the 6202, 6100E, or 6105 controller. The AM6520 protocol uses a specific polling technique, in which each terminal on the system is polled. For example, if 64 terminals share a line, the system sends 64 separate polls even if a TCC is used. In comparison, other protocols make use of a general polling technique, in which a device, known ambiguously as a controller, can support a number of terminals. The line supervisor sends a single poll to the controller (a device controlling the terminals on the line, not an I/O controller) to see if any of the terminals have data to transmit. This cuts down on the number of polls sent on the line and improves response time. The AM6520 protocol, with its specific polling technique, supports fewer terminals per line due to performance implications.

A number of terminals connected with only a single line or cable is a cost advantage. However, the savings must be balanced against performance because the terminal must wait to be polled. Additionally, consider that the time needed for a terminal to be serviced is a factor of the number of devices on the poll list, the data they transmit, and the speed of the line.

AM3270

AM3270 is used exclusively for IBM bisynchronous 3270 terminals (or compatibles). Like AM6520, this I/O process polls the attached terminals and some of the same concerns, such as the performance issues and polling overhead, still apply. However, AM3270 typically uses general polling, in which a group of terminals is polled using only the terminal controller's address (again, the term "controller" is used here to describe a device controlling terminals on a line, such as an IBM 3274, not an I/O controller, such as a 6105).

Typically a controller (as above) can support up to 32 devices (terminals or printers), and all devices need to be attached to the controller (unlike Tandem 652X and 653X terminals and PCs). This reduces the overhead on the line used for polling terminals and allows more terminals per line for the same data traffic. Like AM6520, the devices are identified through the use of a qualified name. The terminals must be defined to the system by way of CMI after every load; this is an additional configuration step. AM3270 uses the 6202, 6100E, or 6105 controller and can support up to 255 devices per process (the practical limit is usually less).

Figure 9

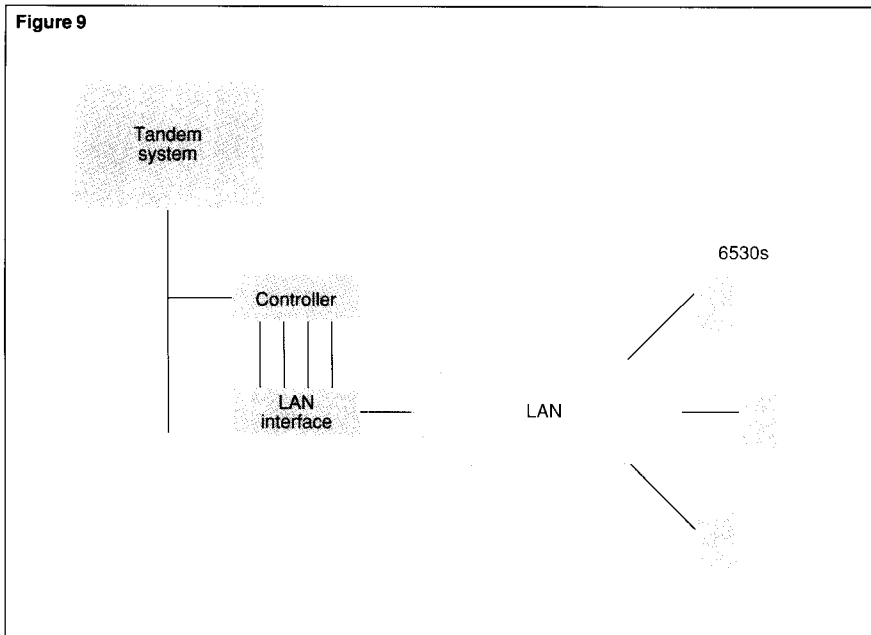


Figure 9.

Ungermann-Bass asynchronous LAN interface. Note that the controller pictured may be any of the Tandem asynchronous controllers, such as the 6303/6304, 6100E sub-system, or 6106.

SNAX/XF

SNAX/XF provides a polling I/O process that connects SNA-type devices to Tandem systems. This protocol uses a general polling technique (which polls only the controller, as above) and, as a result, is more efficient.

SNAX/XF can also be used to connect Tandem terminals through the use of the 6600 cluster controller, which takes the place of an IBM 3274 controller and performs the SNA-to-asynchronous data conversion for Tandem terminals. The 6600 is so similar to an IBM SNA controller that the two controllers can share a multidrop line. Therefore, a SNAX/XF solution may be advantageous when there is a mix of SNA 3270 and Tandem terminals. As in the above solutions, SNAX/XF requires configuration of subunits after a system load with CMI. Terminal names must be qualified. SNAX/XF uses the 6204, 6100E, or 6105 controller; the addressing limit is 255 devices per process.

X25AM

X25AM, Tandem's X.25 protocol, allows any type of asynchronous terminal to be attached to Tandem systems. For the data communication world, X.25 provides long-distance switching at reduced rates. X.25 does not use polling; instead, it functions as a first-in, first-out queue for the most part, although priority packets are expedited through the network. Because either side can act as a supervisor, true full-duplex communication (simultaneous transmission in both directions) is possible.

X25AM can handle up to 255 terminals, each of which is assigned a unique identifier when the call is established so the mixed data traffic can be identified and routed to the assigned subunit for the call. As described for preceding protocols, the subunits must be defined after a system load and terminal names must be qualified. X25AM can use the 6202, 6204, 6100E, or 6105 controller.

Another way of using X25AM is through the use of a packet assembler/disassembler (PAD) to support multiple terminals. A single line can run from the PAD directly into the Tandem system, providing another way to allow multiple terminal access through a single line.

MULTILAN

Tandem's MULTILAN, a set of hardware and software products, allows PC users to link their NETBIOS-compatible⁴ local area networks (LANs) to Tandem systems. MULTILAN allows customers to select from a wide variety of NETBIOS LAN vendors without concern for Tandem compatibility.

In addition to MULTILAN, the LAN technology offered by Ungermann-Bass, a subsidiary of Tandem, allows users to take advantage of a LAN system by using asynchronous ports to LAN converters that are connected to a Tandem system through one of the asynchronous controllers. To the Tandem system, they would appear to be directly connected. LAN technologies allow many devices to share a single line or cable, which can significantly reduce wiring costs and allow a great degree of flexibility in the relocation of terminals within a building. (See Figure 9.)

⁴NETBIOS (Network Basic Input Output System) is a peer-to-peer standard application-programming interface. It is independent of any underlying LAN protocols and therefore allows network applications to be connected to a network as well as ported from one network to another.

Ungermann-Bass also offers an X.25-to-LAN converter. This allows 6530 or any asynchronous terminal to connect through a LAN to a Tandem system by way of X.25. To the Tandem system it appears that these terminals are connecting through an X.25 network. This option connects many terminals to the system but uses a minimum number of ports. The user can take advantage of LAN technology and X.25 multiplexing capabilities for a very cost-effective method of connecting terminals. (See Figure 10.)

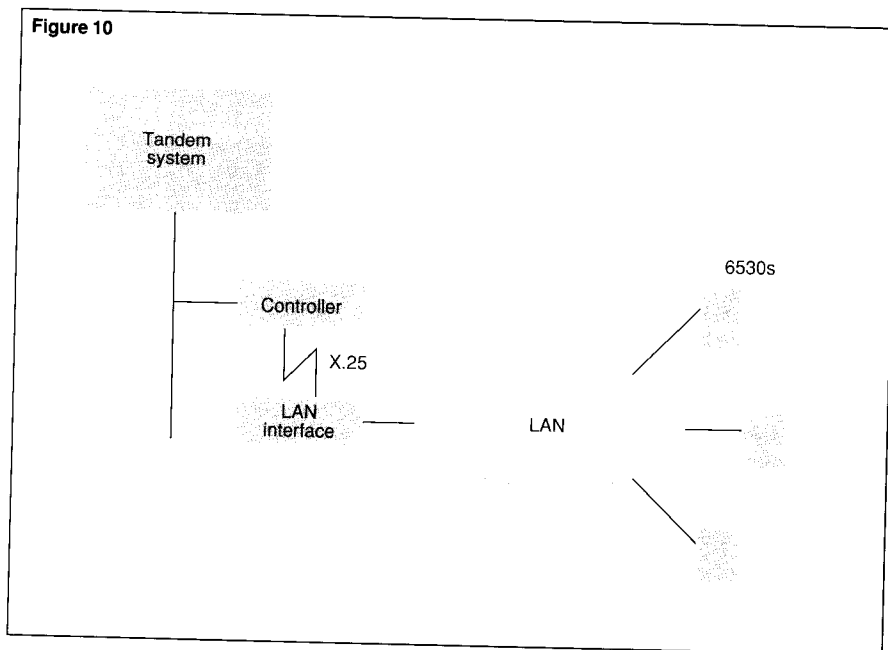
Nonstandard Terminal Connections

There are several alternatives that are available to support terminals that cannot be connected using standard communications software. They include ENVOYACP/XF, SNAX/HLS, and CP6100.

ENVOYACP/XF

ENVOYACP/XF is a bit-synchronous-based protocol and uses the 6204, 6100E, or 6105 controller. Devices conforming to synchronous data link control (SDLC), high-level data link control (HDLC), or advanced data communication control procedure (ADCCP) standards are supported by ENVOYACP/XF, and it is programmed to control the line through GUARDIAN 90 calls and a message control word in the buffer. The ENVOYACP/XF protocol can emulate a supervisory, tributary, or combined station (a combined station is one that has both tributary and supervisory capability, making it inherently full-duplex).

ENVOYACP/XF should only be used in situations where Tandem does not provide high-level support for the required devices. (High-level support refers to an I/O communication process that provides all levels of protocol support, which frees the programmer from the details of the actual line protocol being used.) ENVOYACP/XF is a low-level interface requiring the application programmer to understand the protocol, as GUARDIAN 90 calls (such as READ, WRITE, CONTROL, and SETMODE) affect and control the line. For example, if a bit-synchronous multipoint protocol is being used, the initial READ causes polling to begin; secondary READs get actual data from the line. The programmer also controls the line through the use of the message control word located as the first word in the buffer. For details of this interface, refer to the *ENVOYACP/XF Reference Manual*.



SNAX/HLS

SNAX/HLS (SNA communication services-high-level support) facility is a process that works with SNAX/XF to provide upper layers of the SNA protocol. SNAX/XF provides all the layers for standard 3270 emulation, but only the link layer and some of the path layer for non-3270 devices. With SNAX/HLS, all the layers of SNA are provided. The programmer interfaces to HLS (and therefore to the device) through the use of high-level calls such as OPEN-SESSION and SEND-DATA. This allows the programmers to concentrate on the application rather than on the intricacies of SNA.

Figure 10. Ungermann-Bass X.25 LAN interface. Note that the controller pictured may be any of the Tandem synchronous controllers, such as the 6100E subsystem, 6202, 6204, or 6105.

CP6100

This process only works with the 6100E or 6105 controller. The CP6100 process can allow programmatic control of asynchronous, byte-synchronous, and bit-synchronous protocols. The programmer interfaces to the system through an 8-byte control header, which precedes each request. The requests are sent to the CLIP, which performs tasks based on the requested function within the header. Each request is sent by way of a GUARDIAN WRITEREAD request. The response to the WRITEREAD lets the programmer know the outcome of the request.

As with the ENVOYACP/XF product, CP6100 is a low-level interface and requires that the programmer has knowledge of the protocol. ENVOYACP/XF allows devices not supported by a high-level protocol to be connected through a 6100E or 6105 controller. For details on CP6100 programming, refer to the *CP6100 Programming Manual*.

IDS. Of course, protocol support and actual integration are two different things. A terminal can be connected to the system but still be functionally shut off. PATHWAY, using the Intelligent Device Support (IDS) option, allows a variety of intelligent devices (such as PCs), once connected, to take advantage of the many benefits inherent in the PATHWAY system. IDS assumes that the connected devices are intelligent and do their own formatting, so no attempt is made to format a screen in the standard fashion.

GDS. The Tandem General Device Support (GDS) product specifically addresses terminal connection. Tandem provides a skeleton process that is completed by the customer and communicates to a PATHWAY TCP, making it believe that it is communicating with a Tandem 6530 terminal. The special escape sequences and controls are translated by the completed GDS process so that a foreign terminal can function as a 6530 in a PATHWAY environment. For more information on GDS, refer to the *GDS Programming Manual*.

Sample Configurations

The following samples illustrate the connection of ten 6530 terminals to a Tandem system. Various methods of connection are shown with a discussion of the benefits and concerns of each configuration. The samples use the TERMPROCESS, ATP6100, AM6520, X25AM, and SNAX I/O processes and the 6303/6304, 6202, 6204, 6106, 6105, and 6100E hardware controllers.

TERMPROCESS

TERMPROCESS and the 6303/6304 could be used quite effectively if the terminals are to be located within 1500 feet or less of the system. Each terminal gets a cable running directly to the patch panel. Three I/O slots and 32 channel addresses are lost, but it is possible at a later time to add up to 22 more terminals in this same fashion without further system impact. Any additional cost is incurred for new terminals, cable, and additional patch panels.

ATP6100

The same setup would be available using the 6106 and ATP6100. This configuration needs only a single board (thus only a single I/O slot is needed) and only eight addresses would be taken. Six more terminals can be added without impacting the system (any additional cost would be for terminals and cable).

6100E

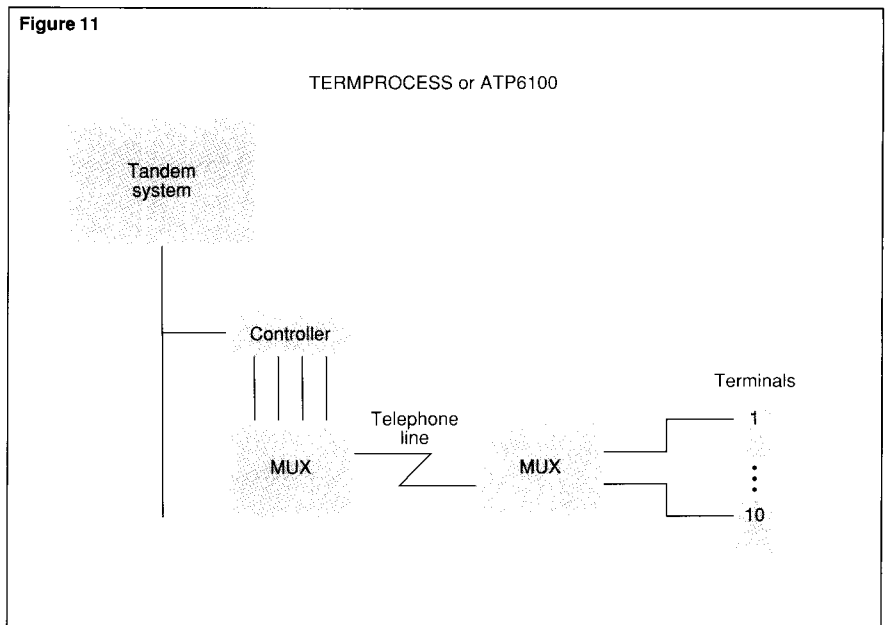
The 6100E subsystem could also be used, but for this number of terminals, it would probably not yet be cost-effective. If the terminal population is expected to grow rapidly, the 6100E should be considered. The 6100E subsystem does not require an extra I/O cabinet and patch panel bays. It runs with a reduced number of CPU cycles (although there would be no reduction in CPU cycles if matched against the 6106) and offers flexible support for a variety of protocols. These features make it possible for the customer to offset or even negate the initial cost of the 6100E.

Multiplexor

In the case where either the terminals are remote or cables cannot be run to each terminal and the terminals are somewhat clustered, any of the above solutions would work along with a device called a multiplexor. A multiplexor is a common device used to share either a telephone line or cable. Through the use of a multiplexor, a single line could be shared by the ten terminals. (See Figure 11.)

Multiplexors can be a very cost-efficient alternative to the "cable per terminal" method of attachment. The price usually includes the multiplexors, built-in modems, and several terminal ports. Additional ports may be added. There are many multiplexor vendors and prices vary. Get a few quotes from these vendors; include the Tandem charges for the controllers, terminals, and cables. Balance the additional cost of the multiplexor against the cost of running a cable to each terminal, if locally attached, or against the telephone company charges to run a dedicated (or switched) line to each terminal.

Locally, the multiplexor may not be cost-effective, but it can quickly pay for itself when used in lieu of separate telephone lines for each terminal. The remote charges include all Tandem costs, multiplexor cost, and telephone company charges for a single dedicated line.



One note about the multiplexor solution is that terminal cabling costs will double if the configuration requires a cable from the patch panel to the local multiplexor and a cable from the remote multiplexor to the terminal. The value of the multiplexor is obtained in either a reduced number of long cables, if local, or reduced number of telephone lines, if remote.

Figure 11.

Multiplexor example. Note that the controller pictured may be any of the Tandem asynchronous controllers, such as the 6100E subsystem, 6303/6304, or 6106. For CLX configurations, the 3606 controller would be used.

Figure 12

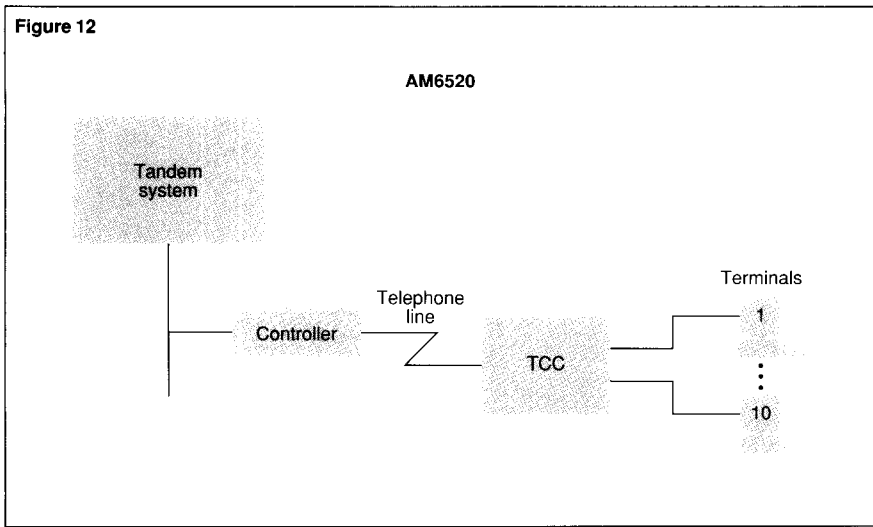


Figure 13

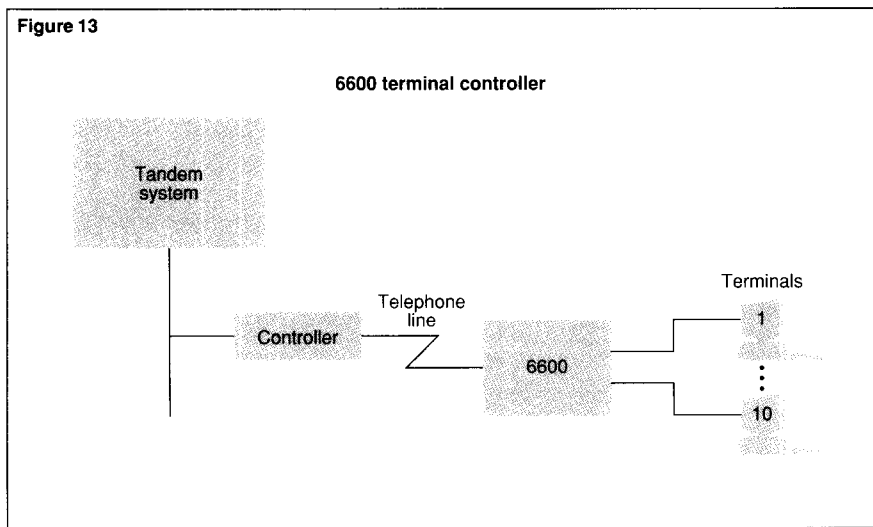


Figure 12. AM6520 and TCC example. Note that the controller pictured may be any of the Tandem byte-synchronous controllers

such as the 6100E subsystem, 6202, or 6105. For CLX configurations, the 3605 controller would be used.

Figure 13. 6600 terminal controller example. Note that the controller pictured may be any of the Tandem bit-synchronous controllers, such as the 6100E subsystem, 6204, or 6105. For CLX configurations, the 3605 controller would be used.

AM6520

Another option for connecting either a local or remote cluster of Tandem terminals is the AM6520 process. This is most often used in conjunction with a TCC, which has eight ports. Each port is capable of supporting up to eight daisy-chained terminals. The TCC in conjunction with the AM6520 protocol provides a synchronous-to-asynchronous protocol conversion and allows a number of terminals to share a single line. AM6520 has the advantage of having all ten terminals use only a single port on a controller (dependent on application and response-time requirements).

Currently, the TCC is substantially cheaper than a multiplexor. Although this makes it a more cost-effective solution, the response time is often not as good. Besides saving on the TCC itself, only one set of cables is needed from the TCC to the terminals, which means additional savings are achieved. If used with the 6202 controller, this configuration has some CPU impact, as an interrupt is generated by the controller for every pass through the poll list. If used with the 6105 or 6100E system, this is not a concern. (See Figure 12.)

It should be noted that Tandem considers the AM6520 protocol and the TCC mature products, which means that no new features will be offered. The preferred approach is through the use of SNAX 3270 and the 6600 terminal controller, which implements a better polling protocol.

6600 Terminal Controller and SNAX 3270

Another possible solution is the 6600 terminal controller (6600) that, in the IBM 3274 sense, connects Tandem terminals into the system using the SNAX 3270 protocol. The 6600 makes use, as mentioned previously, of a general polling technique, a more efficient polling method than the specific polling technique used by AM6520. (See Figure 13.)

The base 6600 system comes with 4 ports and is expandable to 12 ports (by adding two four-port options). It is approximately the same cost as the multiplexor. The advantages are that only a single cable for each terminal is needed (from the 6600 to the terminal), only a single device is necessary (one 6600 versus a pair of multiplexors), an efficient protocol is used (no excess interrupts when using the 6105 or 6100E system as compared with the 6204), and the base 6600 system is a single-vendor solution. Anyone familiar with debugging communication problems will appreciate the last advantage.

X.25

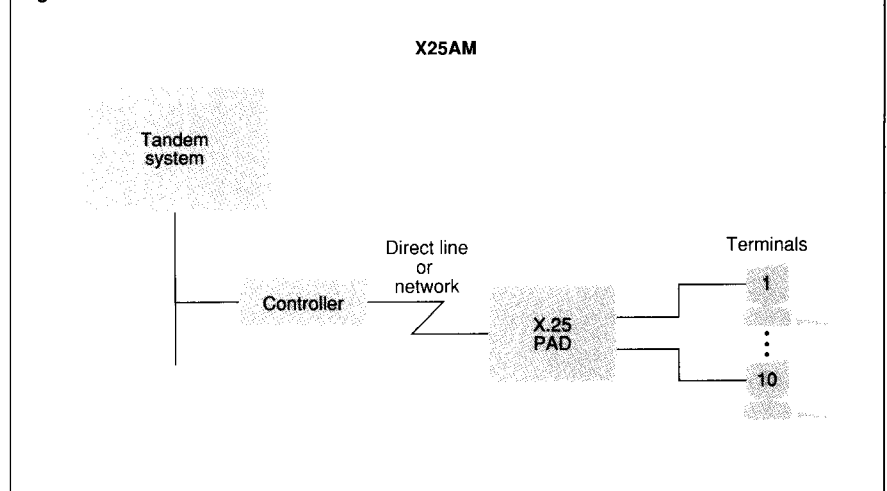
The last solution involves the use of X.25, which can be the most cost-effective solution if the ten terminals are not local and are not clustered; that is, they are distributed around the country (or world). X.25 networks, also known as value-added networks or public data networks (PDN), are widely used in Europe where telephone company charges prohibit the use of leased facilities. Terminals make a local call into the network and give the address of the host system with which they want to communicate. The network then sets up what is known as a virtual circuit, and the terminal is connected just as if it dialed in directly. The terminal actually calls a PAD, which provides the services of connection, monitoring, and asynchronous-to-X.25 protocol conversion.

The networks charge for the local line from their network to the host. Check with the PDN for their charges, which should include the line, modems, and all service on the line. The networks also charge for connection time and data transmitted. The Tandem system can be configured to connect to an X.25 network. At a certain point, it may be more advantageous to buy an X.25 PAD and run it directly into the Tandem system. X.25 PADs generally cost about as much as a multiplexor or 6600. The configuration is quite similar. (See Figure 14.)

Conclusion

Tandem offers a range of terminal connection alternatives that provide the flexibility Tandem customers require when adding terminals to Tandem systems. An optimal terminal configuration is established by examining all the requirements for the terminal connections and matching them with the features offered by Tandem's communications controllers and software. The many possibilities offer the flexibility, cost-effectiveness, and growth that is necessary in today's rapidly changing business environment.

Figure 14



References

- Coyle, A. Introduction to MULTILAN. *Tandem Systems Review*. Vol. 4, No. 1. Tandem Computers Incorporated. Part no. 11078.
- Dynamic System Configuration Manual*. Part no. 84167. Tandem Computers Incorporated.
- System Generation Manual, Vol. 2*. Part no. 84095. Tandem Computers Incorporated.
- 653X Multi-Page Terminal Programmer's Manual*. Part no. 82310. Tandem Computers Incorporated.
- ENVOYACP/XF Bit-Oriented Protocols with Extended Functions Reference Manual*. Part no. 84116. Tandem Computers Incorporated.
- CP6100 I/O Process Programming Manual*. Part no. 82410. Tandem Computers Incorporated.
- General Device Support (GDS) Programming Manual*. Part no. 84129. Tandem Computers Incorporated.

Figure 14.

X.25 and PAD example. Note that the controller pictured may be any of the Tandem bit- or byte-synchronous controllers that depend on the PDN, such as the 6100E subsystem, 6202, 6204, or 6105. For CLX configurations, the 3605 controller would be used.

Justin Simonds is an advisory analyst on the technical support staff of the Western Region. He has also worked as an account analyst in both Analyst and Customer Education. Before joining Tandem he was a senior system engineer for a major U.S. public data network.

NonStop CLX: Optimized for Distributed On-line Transaction Processing

The proliferation of microprocessors in the workplace has created an entirely new set of transaction driving devices, including personal computers, cash registers, automated teller machines, point-of-sale terminals, and even gas pumps. The result of this increased transaction demand is a need for processing that is geographically distributed. Distribution permits greater parallel processing and a structuring of computer resources that matches the hierarchical structure of most businesses and most business transactions. The NonStop CLX™ system was designed to meet on-line transaction processing (OLTP) needs and, in particular, the needs of distributed processing.

Distributed OLTP Requirements

OLTP has more stringent requirements than traditional data processing. These include high availability, data integrity, modular growth, fast response time, and flexible connectivity.

Distributed OLTP has additional criteria. Most important is lower cost of ownership, including costs associated with the maintenance and operation of a system, as well as the initial purchase price. Lower initial cost is achieved by more integrated packaging and fewer customer environmental requirements (e.g., power and cooling). Lower cost of operation depends on simplified human interfaces, remote support, and reduced service costs.

Distributed OLTP also requires the ability to network systems and support distributed databases. Networking is essential to handle transactions that are remote to the distributed node and to allow the management of the set of distributed systems from a central site. Tandem hardware and software already support networks and distributed databases, but this need becomes more critical as the level of distribution increases.

NonStop CLX Hardware Design

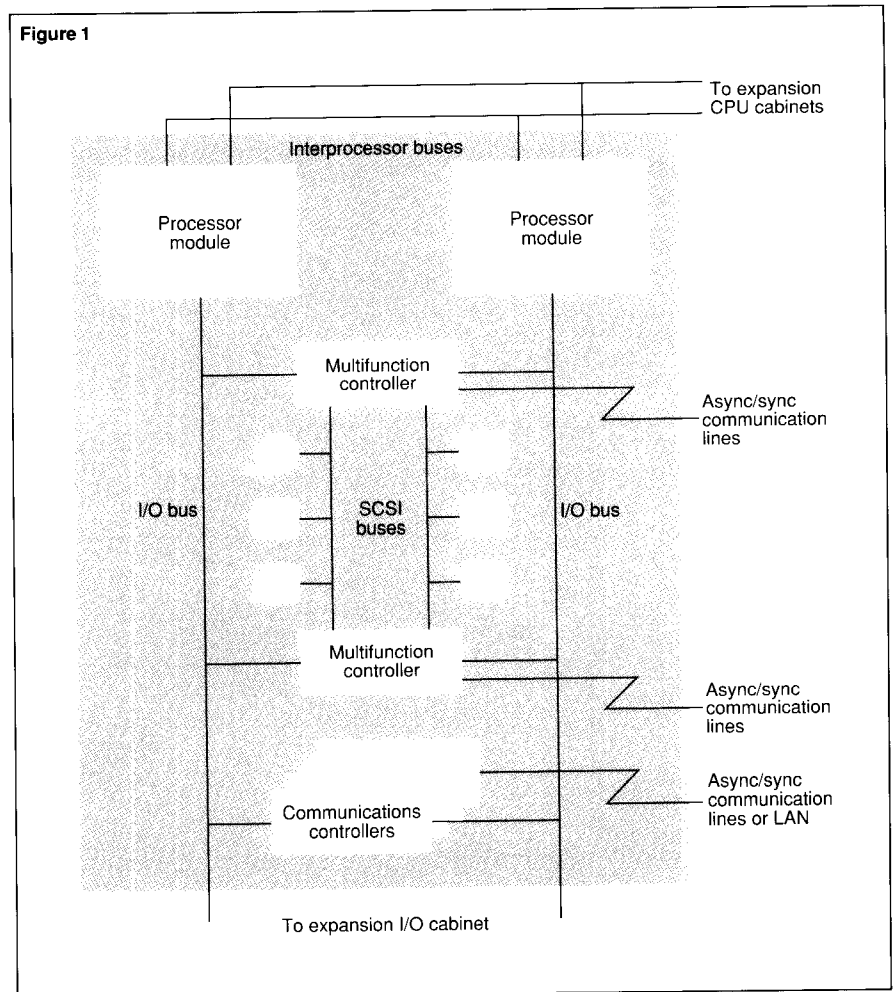
Tandem's NonStop TXP™ and VLX™ systems (Bartlett, Gray, and Horst, 1987) were optimized to handle the medium and large transaction requirements of central or regional business centers. The primary goals of these systems were high performance and maximum flexibility. The NonStop EXT™ line was Tandem's first move to lower the cost of system ownership and to bring the processing of transactions closer to their generation point. It was realized, however, that to reduce these costs dramatically, an entirely new system was needed. The CLX is a completely new implementation of the NonStop architecture, yet maintains full software compatibility with the TXP, VLX, and EXT systems.

The primary technical goals of the CLX were increased integration and ease of service. Integration in this case refers to a reduction in the number of boards, connectors, and components used in the system. This in turn reduces system cost, power consumption, and air conditioning requirements, allowing the CLX to be operated in an office environment.

Ease of service is aided by integration but also requires the use of advanced packaging techniques and software support. In the resulting CLX system, 80% of system components can be serviced by a customer. Programmatic fault-isolation and repair techniques allow 98% of all system incidents to be user-serviced. A CLX can be user-installed or upgraded. Alternately, the user can choose full Tandem support, but cooperative maintenance greatly reduces overall maintenance costs.

The base configuration CLX block diagram shown in Figure 1 is very similar to the traditional Tandem NonStop system block diagram. The principal difference at this level is that all of these elements are contained in a single enclosure the size of a four-drawer file cabinet. This includes:

- Up to two CPUs with expansion memory.
- Two multifunction controllers.
- Four communications controllers.
- Five 300-Mbyte disk drives.
- One cartridge tape drive.



A system can also be expanded with up to two additional CPU cabinets, each having an attached I/O cabinet.

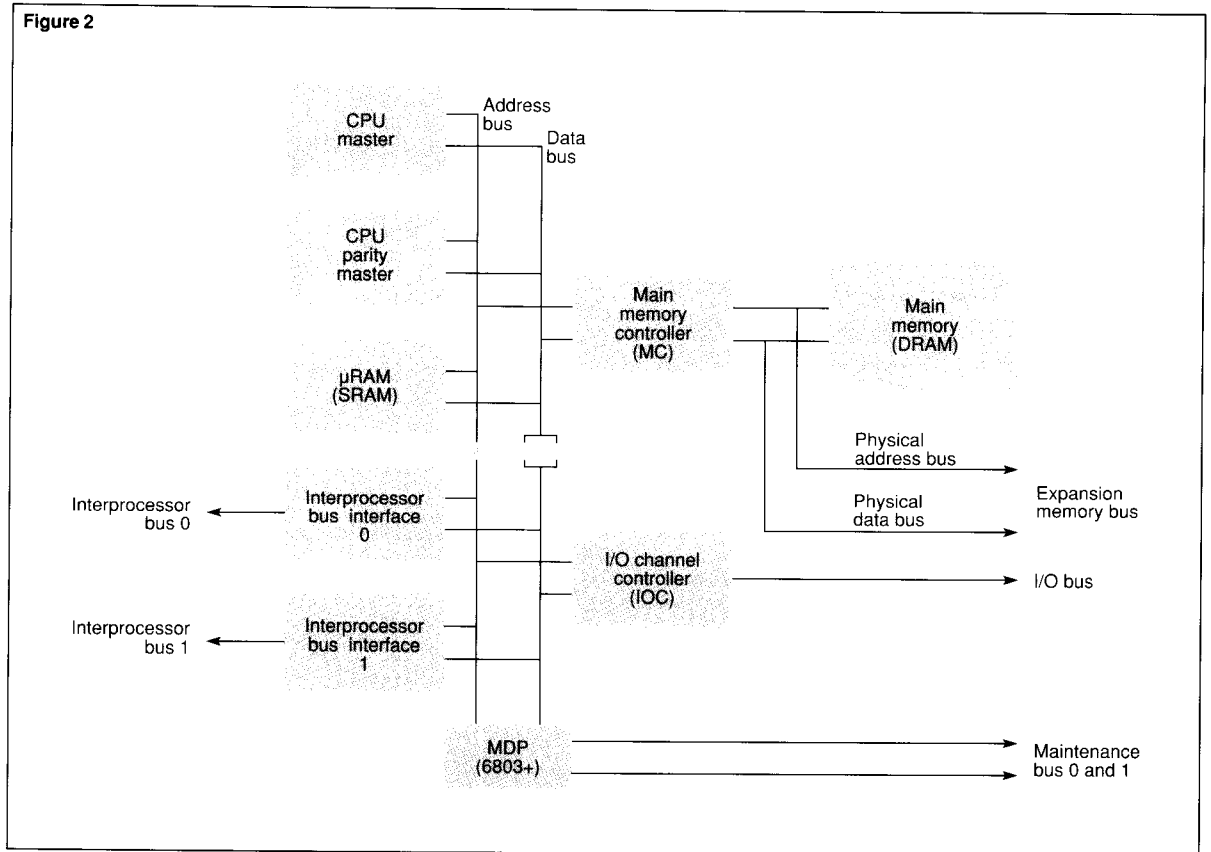
Processor Subsystem

The new CLX processor reduces the entire instruction processing unit and base main memory to a single printed circuit board (PCB). By contrast, a base VLX has three boards, and the base TXP has four. The CLX CPU board contains six application-specific integrated circuits (ASICs) that implement the majority of the processor logic.

Figure 1.
Base cabinet block diagram. A complete CLX system is packaged in a cabinet that is roughly the size of a four-drawer file cabinet.

Figure 2.

CLX processor architecture. The single μ RAM array encompasses the function of microcode control store, page translation cache, and memory cache.



Technology. Power and integration constraints dictated the use of complementary metal oxide semiconductor (CMOS) logic instead of the bipolar technology used in other Tandem CPUs. A silicon compilation design methodology (Stenzel, 1988; Johnson, 1984) was used to implement the ASICs because it permitted a shorter design cycle than hand-crafted integrated circuit design and higher logic density than other ASIC techniques (e.g., gate arrays or standard cells).

Silicon compilation begins with textual descriptions of the required logic blocks, which are then “compiled” into full custom layouts of the transistors needed to realize each block. The user then places these blocks relative to one another and specifies their interconnections. From this, the compiler determines the topology for the necessary wires and a final geometric database that can be used to fabricate the integrated circuit.

The processor also used the latest RAM technology for the high-speed static RAMs (16 Kbit \times 4) and main memory dynamic RAMs (1 Mbit \times 1). Combined with the CPU architecture, this permitted more than a fourfold reduction in the number of RAM parts used in previous NonStop processor designs.

Architecture. The architecture of the CLX CPU is a hybrid of the very parallel structures found in Tandem's larger VLX and TXP CPUs, together with the structures found in the highly integrated microprocessors. (See Figure 2.) This joining of techniques yields high performance as well as a high level of integration.

The internal structure of the CPU chip is parallel. Each clock cycle initiates execution of a microcode instruction word that can independently perform an ALU¹ operation, a conditional branch, a memory address calculation, and an external bus operation. Microcode execution is pipelined across three clock cycles, providing another dimension of parallelism. External to the CPU, there is a single address and data bus over which the CPU communicates with the microcode control store, memory cache, main memory, and I/O interfaces. A single bus structure allowed the integration of the CPU into one chip but was potentially a major bottleneck. This was avoided by supplementing the external microcode with a small on-chip microcode ROM and by carefully defining the operations supported by the other functional units on the bus.

The ROM microcode implements common sequences such as instruction prefetch, main memory write-through, and the inner loops of block memory and I/O transfers. Other ROM locations are dedicated to frequently used instructions. The resulting architecture has a slower cycle time than other Tandem CPUs, but compares favorably on a microcycle basis with TXP and VLX systems for many of the common instruction types. (See Table 1.)

The rest of the processor function is implemented in three additional custom integrated circuit types:

- The main memory controller (MC).
- The interprocessor bus interface (IPB).
- The I/O channel controller (IOC).

¹Arithmetic Logical Unit.

Table 1.

The microcycle counts for simple instructions on CLX compare favorably with the larger TXP and VLX processors. The clock cycles themselves, however, are longer.

	Clock cycles		
	CLX	TXP	VLX
Register stack instruction	2	2	1
Memory to stack instruction	3	3	2
Stack to memory instruction	5	4	3
Branch instruction (taken/not)	3/3	4.5/3	3/2
Microcycle time (ns)	133	83	83

The MC controls the 4 Mbytes of on-board DRAM and the 2 to 8 Mbytes of expansion DRAM. This chip contains logic to do full single-bit error correction and double-bit error detection, as well as buffers and control logic to optimize the CPU's interface to main memory. The IPB chip is replicated with each integrated circuit controlling one of the two interprocessor buses. These parts contain 16-word input and output queues that allow packet transfers between processors without CPU intervention. Finally, the IOC chip controls the I/O bus that connects the processor to one of the two ports on each controller board.

The final section of the processor is a single-chip microcomputer (Motorola 6803), which adds a diagnostic interface to the CPU through dual maintenance buses.

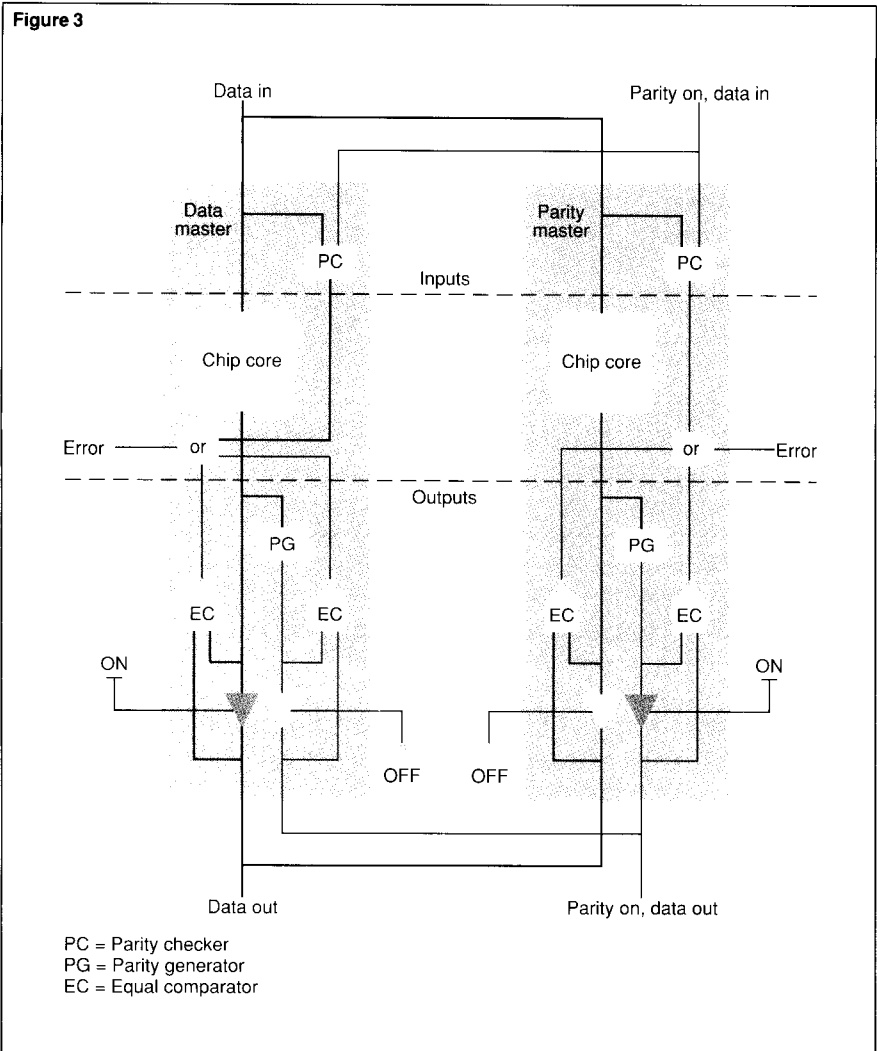


Figure 3.
The CLX CPU chips are cross-coupled to provide a high degree of data integrity.

Data Integrity. OLTP requires that the integrity of the database not be compromised. This requirement becomes more acute in a fault-tolerant system, which must isolate faulty modules before they can corrupt their backup and cause a system failure.

Tandem CPUs use various techniques to assure data integrity. Some of these techniques include:

- Parity maintenance and prediction together with parity checking.
- Duplicate and compare.
- Error correcting codes.
- End-to-end checksums.
- Cyclic redundancy checks.

Duplicate and compare has been used at Tandem primarily within I/O controllers, but the level of integration and the design methodology of the CLX processor made it attractive to use on the CPU chip. The use of full-custom logic allowed for the comparison logic to be built into the I/O pads of the CPU chip itself.

Parity encoding is used to cover transfers between the CPU chip and the other parts of the processor. The inherent redundancy of the parity lines allows cross coupling of the checking logic in the CPUs. (See Figure 3.) The duplicate CPU chip is not simply a slave to the master chip but is responsible for driving the parity lines that cover the data bits. Any internal failure of the data master will be caught by the parity master checking the data lines. Any failure of the parity master, including latent faults in checking logic, will be caught when the master chip checks the parity lines.

Multifunction Controller/Storage Subsystem

While integration of the CLX processor required a new implementation architecture, integration of the I/O subsystem required an entirely new systems approach to the design. The result is a single controller serving multiple functions (disk, tape, communications, and maintenance) and intelligent disk and tape units that are controlled on a pair of common buses.

Advances in both hardware and software were required to realize the multifunction controller (MFC). One such innovation was the use of the small computer systems interface (SCSI) to communicate to the disk and tape drives. The intelligence of the SCSI disk and tape drives simplified the real-time control requirements of the MFC, and removed the need for dedicated disk and tape interfaces. Fault tolerance was added to the disk subsystem by providing each MFC with an interface to two SCSI buses and distributing primary and mirror disk drives on opposite buses. The use of 5¼-inch disk and tape devices permitted integration of these units (with up to 1.5 Gbytes of disk storage) in the base cabinet. On the software side, a significant amount of controller software was developed along with a real-time executive that coordinates the multi-threaded activities of the MFC.

Figure 4

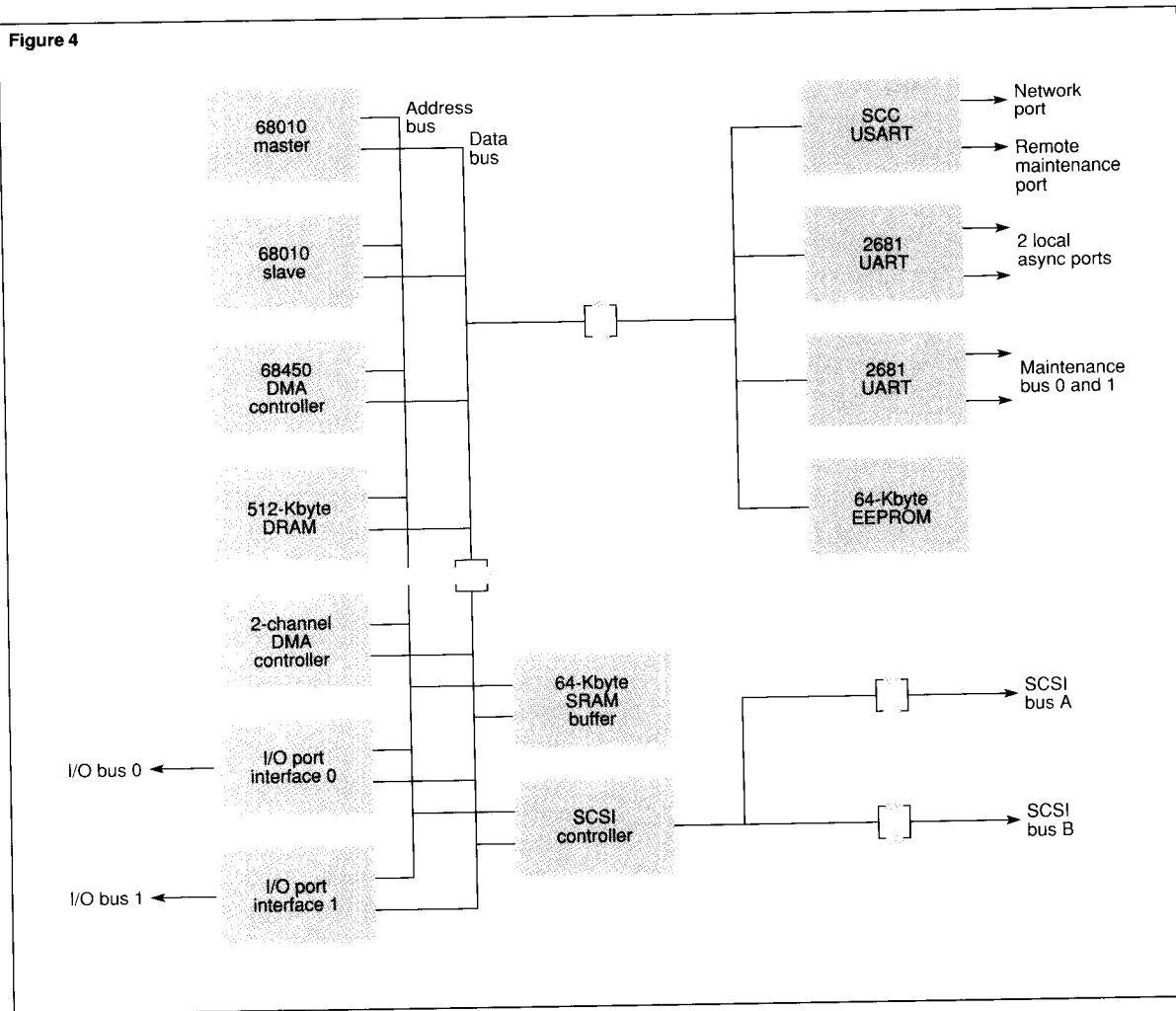


Figure 4.
MFC architecture.

As with the CLX processor, the integration used in the MFC could potentially compromise performance. This was avoided through technology and architecture. The technology included Motorola 68010 microprocessors (duplicated for data integrity) and CMOS gate arrays. The MFC architecture was based on a dual bus structure developed for Tandem's highest performance disk controller (3120). This architecture uses a dedicated static RAM and DMA² controller on their own bus to buffer disk and tape data through the MFC without the microprocessor becoming a bottleneck. (See Figure 4.)

The communication and maintenance functions of the MFC complete the requirements of a minimal CLX system. Each controller includes two asynchronous ports for local communications and a synchronous network connection to support distributed operation. Each MFC also interfaces to the internal maintenance buses and the external remote maintenance interface. The external interface connects to an asynchronous modem to provide remote support of the system.

²Direct Memory Access.

Communications Subsystem

Most OLTP applications require more communication interfaces than the MFC supports. The base CLX system permits up to four additional communications controllers in the base cabinet. Tandem's existing line of communications controllers met most of CLX requirements except support of user service. Separating the external line interface logic from the controller and mounting it on a card that plugs in directly opposite the main controller on the other side of the backplane reduced the need for a large amount of internal cabling. Based on a principle first used on Tandem's dedicated communications subsystem cabinet (6100), these backplane interconnection cards (BICs) remove all of the internal cabling while still supporting a variety of electrical interfaces on the communication lines. The concept was extended further in the CLX architecture to include system interfaces such as the interprocessor bus controllers and cross-cabinet bus extender cards.

User Interface and Packaging

Reducing cost of ownership through user service was an important consideration in packaging the CLX system. All customer-replaceable units (CRUs) had to be easy to identify, remove, handle, and reinstall. The entire system had to fit into the office environment and have a non-threatening appearance inside and out.

Using BICs eliminated the need for cables and patch panels, but additional care was taken to ensure that no electrical or mechanical safety hazards were present in all user-serviceable sections of the system. All disk and tape CRUs were packaged as plug-in modules, as were the power supplies and battery backup modules. A system control panel was added to the front of the CLX to permit system start-up, shutdown, and other control and monitor functions.

Maintenance Subsystem

As previously stated, dual maintenance buses run throughout the system and connect the CPUs, MFCs, power monitor modules (PMMs), and the system control panel (SCP). These buses provide a diagnostic and maintenance interface to these subsystems and are controlled by a pair of MFCs responsible for forwarding relevant system events for analysis by maintenance software. These events include the failure of the CPU, MFC, or power supply; an under-voltage from a power supply; or an over-temperature indication. The MFCs controlling the maintenance buses are referred to as the remote maintenance interfaces (RMIs) because they are also responsible for the interface to the external remote maintenance port.

NonStop CLX Software Design

The CLX had to provide full software compatibility with the larger Tandem systems. Distributed systems require the same full-function OLTP software as their larger counterparts, and networks of systems have an even more pressing need for such intersystem compatibility. Software supported on the CLX is identical to the VLX, TXP, and EXT machines.

In addition to the microcode and firmware needed to support the new CPU and MFC, the principal changes to the CLX system software are the enhancements to user service and remote support. Both of these functions work together with the existing Tandem Maintenance and Diagnostic Subsystem (TMDS). TMDS software interacts with low-level monitoring and diagnostic software to locate, diagnose, and guide the user to on-line replacement of failing modules. TMDS software also interacts with the RMI software to dial out through the MFC maintenance port to log events at a Tandem On-line Support Center (OSC). The OSC can analyze the event, dial back into the system if necessary, and work with the user to rectify the problem. These dial-in and dial-out facilities are password-protected and can be enabled to disabled as the customer sees fit.

Both local and remote maintenance functions are enhanced by the fault-tolerant nature of the CLX because the power of full-function system software can be used in performing maintenance and repair activity.

System Performance

As with larger Tandem systems, support for high performance and scalability of performance are required of the CLX. An additional requirement is to support a very minimal cost node in order to allow maximum distribution of the processing power.

A single-cabinet CLX supports a base fault-tolerant configuration and a reasonable range of storage and communication interfaces. This configuration supports 5 transactions per second (tps)¹ on a fully functional SQL-coded version of the standard debit-credit benchmark (Anon, et al., 1985).

While the base cabinet can handle average system requirements, the CLX supports both expansion and contraction from this configuration. Up to three CPU cabinets can be cabled together to form a six-processor system with 15 debit-credit tps performance. For I/O-intensive applications, each CPU cabinet can be supplemented with an additional I/O cabinet that supports up to 1.8 Gbytes of disk storage and eight additional communication controllers.

To support very small nodes, the CLX can be configured as a single-processor system. While this configuration does compromise fault tolerance, it supports data integrity and yields a minimal system cost. The single-processor system is packaged in the same base CLX cabinet, allowing expansion to two or more processors if performance or capacity requirements grow.

Conclusion

Design of an OLTP system optimized for distributed processing required contributions from every area of system design. The resulting CLX system provides unequaled distributed transaction processing power and price/performance. The CLX architecture and packaging are the basis of increasingly powerful, yet easy to maintain, computer systems.

References

- Anon, et al. 1985. A Measure of Transaction Processing Power. *Datamation*. Vol. 31, No. 7.
- Bartlett, J., Gray, J., and Horst, R. 1987. Fault-Tolerance in Tandem Computer Systems. *The Evolution of Fault-Tolerant Computing*. Vol. 1.
- Johnson, S. 1984. Silicon Compiler Lets System Makers Design Their Own VLSI Chips. *Electronic Design*. Vol. 32, No. 20.
- Lenoski, D. 1988. A Highly Integrated, Fault-Tolerant Minicomputer: The NonStop CLX. IEEE COMPCON '87.
- Stenzel, W. 1988. Silicon Compilers Tame 10,000-gate-plus ASICs, Gate Arrays. *EDN*. Vol. 33, No. 9.
- Zwagerman, J. 1988. High-Level Integration Brings NonStop CLX System to the Office. *Tandem Update*. Vol. 88, No. 2.

Acknowledgments

Thanks are due to the numerous hardware and software engineers and all the others who helped design and build the CLX system. Thanks also to the reviewers who added greatly to this article, especially Jay Zwagerman for his comments on CLX system issues.

Dan Lenoski has worked in processor development at Tandem for over five years, most recently as technical lead for CLX processor development. He received a B.S.E.E. from the California Institute of Technology and an M.S.E.E. from Stanford University, and is currently a Ph.D. candidate in Electrical Engineering at Stanford.

¹All rates are quoted for 2-second response time for 90% of all transactions.

NetBatch: Managing Batch Processing on Tandem Systems

Batch processing continues to play an important part in the support of on-line transaction processing (OLTP). In response to its customers' batch processing requirements, Tandem developed NetBatch™-Plus, a software product that allows users to maximize the throughput of jobs by making the most effective use of Tandem's multiprocessor architecture. In addition, NetBatch-Plus reduces human resource requirements by automatically running the batch programs without operator intervention.

NetBatch-Plus is composed of two integrated products: NetBatch, which was introduced with GUARDIAN® 90 operating system, version C00, and the NetBatch Screen Interface, which will be available with the release of GUARDIAN 90, version C20. (See Figure 1.)

This article describes how NetBatch answers the requirements of a batch processing management product. It assumes that the reader is familiar with NetBatch, version C10, terminology and functions.

Improving Batch Processing

In a traditional batch processing environment, sequential processing of batch work is handled by one operator at a single terminal. In contrast, Tandem's NetBatch-Plus uses Tandem systems' parallel architecture to automatically process the jobs concurrently on different processors. NetBatch-Plus manages the execution of batch jobs based on scheduling options, distributes the execution of jobs across different processors, controls all processes started by a job, and maintains dependencies between jobs. Once the options are selected on the NetBatch-Plus screens, all batch processing takes place without operator intervention.

NetBatch-Plus is also a template around which batch applications can be developed. It provides guidelines for operational solutions so that the application can make the most productive use of the available hardware, software processes, and human resources. For example, NetBatch-Plus takes advantage of Tandem systems' parallel processing by allowing operators to specify the CPU and GUARDIAN 90 priority for multiple batch processing jobs initiated by multiple users.

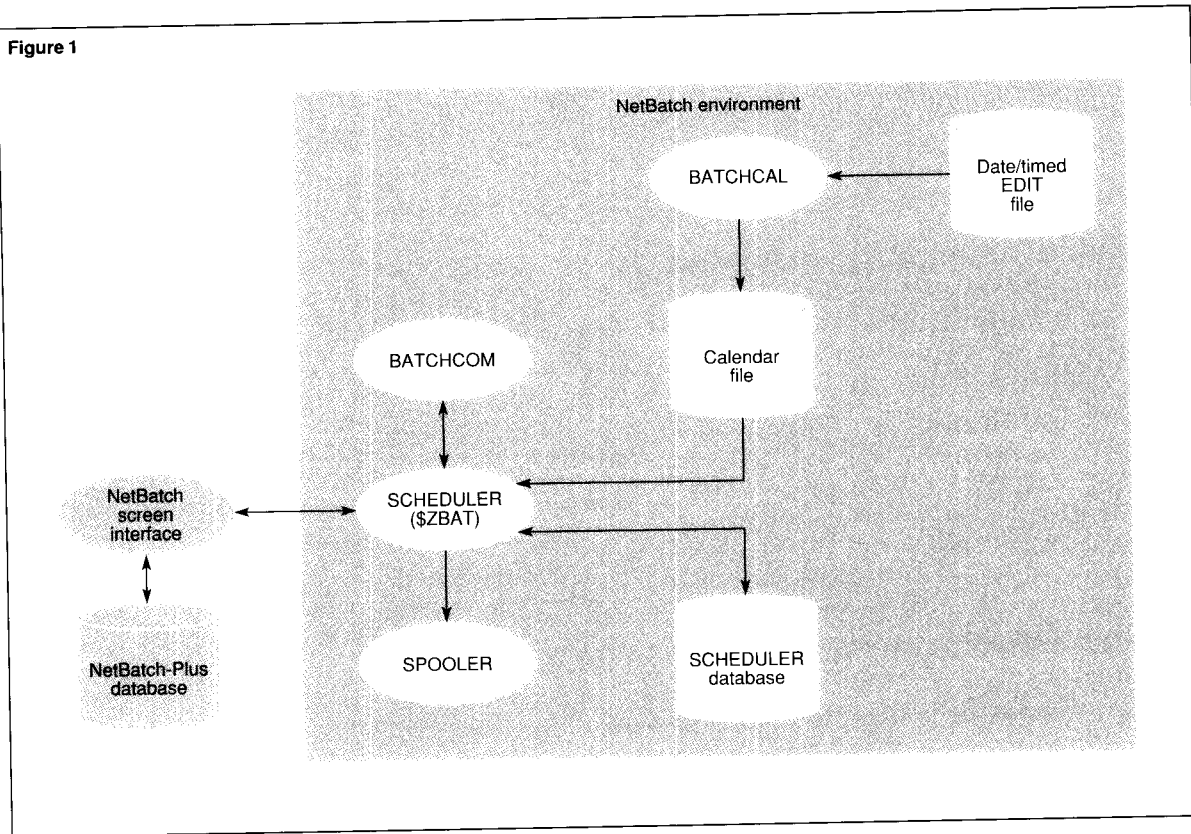
NetBatch Modules

The product NetBatch (without the Plus enhancement) is made up of three components: BATCHCOM, BATCHCAL, and SCHEDULER.

BATCHCOM. BATCHCOM is a conversational interface to the SCHEDULER that allows a user to submit, monitor, and control jobs throughout a distributed environment.

Figure 1

Figure 1.
NetBatch components.



BATCHCAL. BATCHCAL is a program that creates calendar files for scheduling recurrent jobs with complex starting times. An EDIT file that contains either specific or generic dates (such as the last day of the month) is used to invoke the BATCHCAL program. The output is a formatted file that is used by the SCHEDULER.

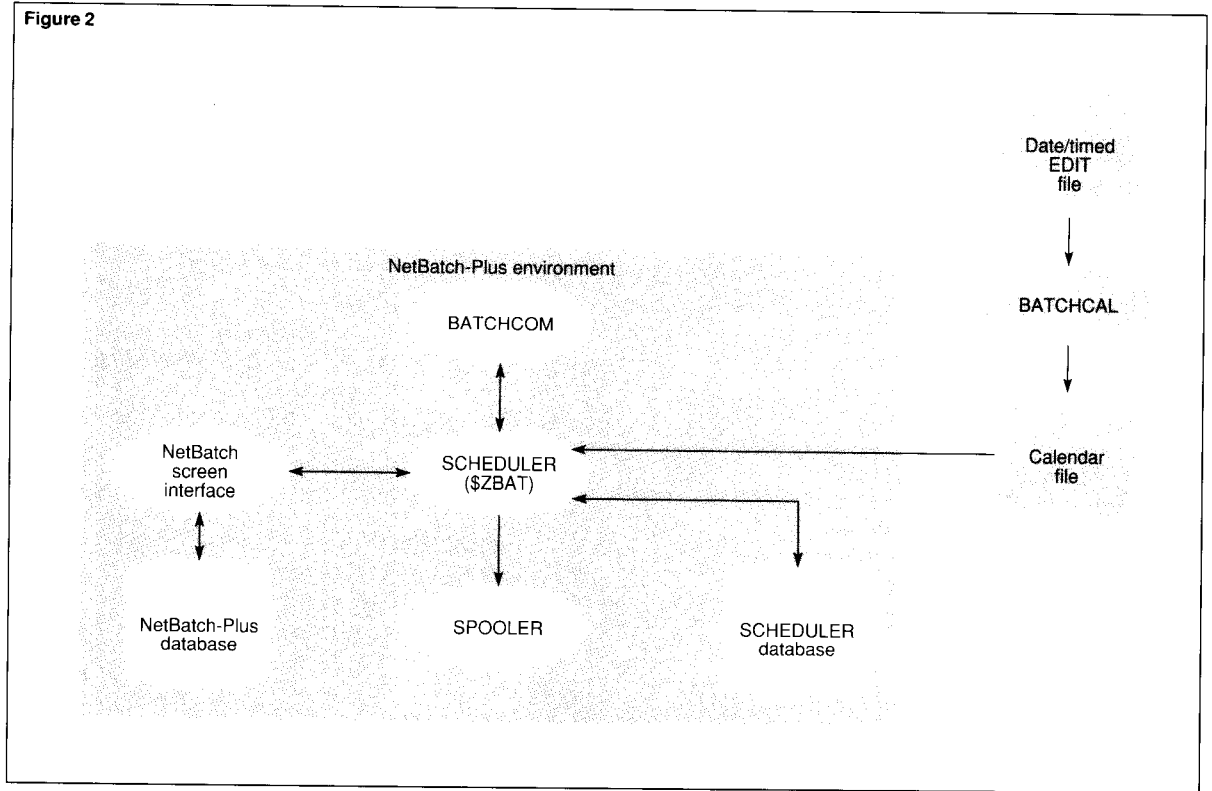
SCHEDULER. The SCHEDULER is a fault-tolerant, process-pair server that maintains supervisory access to all batch processing components. It maintains a queue of all job requests prior to execution, selects those jobs for execution based on each job's scheduling criteria, controls the number of concurrently running jobs on the system, and maintains a central log of all events associated with the execution of each job. Each SCHEDULER process maintains the relationship between its objects. (See Figure 1.)

A NetBatch *job* is a collection of one or more processes. The NetBatch object JOBCLASS is a logical grouping of related jobs. By linking JOBCLASS to an EXECUTOR object, jobs belonging to the JOBCLASS are submitted together to specified CPU(s). A JOBCLASS may belong to several EXECUTORS.

An EXECUTOR object links a JOBCLASS (and ultimately jobs) to a specific CPU. It starts the initial job process to execute in this specific CPU. Each EXECUTOR can have from one to eight JOBCLASSES assigned to it.

It is important to note that the EXECUTOR program is not the same as the EXECUTOR described above. The EXECUTOR program is the object program that executes each command line in a job's IN file. It can be a command interpreter (such as COMINT or TACL™, the Tandem Advanced Command Language), a program (such as PUP, the peripheral utility program), a language compiler (such as COBOL or TAL, the Transaction Application Language), or the MIS-BATCH execution process (BPROC). In any case, the commands in the job's IN file must conform to that of the EXECUTOR program. By default, the SCHEDULER's EXECUTOR program is TACL.

Figure 2.
NetBatch-Plus
 components.



NetBatch Screen Interface

In addition to the NetBatch components described above, NetBatch-Plus includes a block-mode screen application that interfaces with the NetBatch SCHEDULER. NetBatch-Plus records in a database job descriptions, dependency, and selection criteria. Jobs are selected for submission to NetBatch individually or in bulk (such as jobs submitted as a payroll application).

NetBatch-Plus provides a number of displays in tabular format of NetBatch's status and interfaces into other Tandem subsystems. System management and operation of batch applications throughout a network is facilitated by powerful defaulting mechanisms and global catalogs of the GUARDIAN 90 parameters, PARAM, ASSIGN, and DEFINE. (See Figure 2.)

NetBatch Features

NetBatch as described above provides the following features:

- Balance of the batch processing load through resource allocation.
- Control of the impact of batch processing.
- Job scheduling and submission.
- Batch processing visibility and control.
- Audit trail of activities.

Balance of the Batch Processing Load through Resource Allocation

NetBatch uses the multiprocessor Tandem architecture to maximize the overall throughput of batch processing. The Tandem system manager, knowing the historical performance of all applications, determines how many jobs, JOBCLASSES, EXECUTORS, and SCHEDULERS will be executing.

Each JOBCLASS is assigned to an EXECUTOR, although a JOBCLASS may belong to other EXECUTORS as well. Because each EXECUTOR must have a CPU associated with it, the system manager determines which CPU the EXECUTOR will run in. All JOBCLASSES and associated jobs running under the EXECUTOR will run in the EXECUTOR's designated CPU. In this manner, the system manager can ensure that jobs of a given JOBCLASS are run in the preferred CPU(s).

The example in Figure 3 shows three JOBCLASSES and their assigned EXECUTORS operating in a three-processor Tandem system:

- JOBCLASS compile is associated with EXECUTOR EX-1 TACL. EXECUTOR EX-1 is assigned to CPU 1. (All jobs from JOBCLASS COMPILE will run only in this CPU, and the EXECUTOR orders jobs sequentially.)
- JOBCLASS default is associated with EXECUTOR EX-2 TACL. EXECUTOR EX-2 is assigned to CPU 0.
- JOBCLASS production is associated with EXECUTOR EX-2 TACL and EXECUTOR EX-3 TACL. EXECUTOR EX-2 is assigned to CPU 0 and EXECUTOR EX-3 to CPU 2. (With two JOBCLASSES to select from, the SCHEDULER selects a job for execution based on the order in which the JOBCLASSES were assigned to the EXECUTOR.)

When jobs are ready to execute, an initial process is started in the EXECUTOR's CPU to execute the RUN commands in the specified CPUs. At this level there are two exceptions to the CPU resource allocation scheme that are beyond the control of the SCHEDULER:

- Within the job IN file, a RUN command designates a CPU, such as
 RUN GLSORT /CPU 1/
 RUN GLMERGE /CPU 2/.
- If the EXECUTOR program is COMINT or TACL, \$CMON (command interpreter monitor process) may determine on which CPU the execution of a process takes place (if not specified in the RUN command).

Figure 3

	JOBCLASS compile	JOBCLASS default	JOBCLASS production
EXECUTOR EX-3 \$SYSTEM.SYSTEM.TACL			CPU 2 CPU 1 CPU 0
EXECUTOR EX-2 \$SYSTEM.SYSTEM.TACL		CPU 2 CPU 1 CPU 0	CPU 2 CPU 1 CPU 0
EXECUTOR EX-1 \$SYSTEM.SYSTEM.TAL	CPU 2 CPU 1 CPU 0		

The recommended method is to remove all processor designations from within the job IN file and have the EXECUTOR control the throughput of job submission as well as the CPU in which they will run. This will include the removal of process CPU designation determined by \$CMON, which may be accomplished by modifying the \$CMON code to ignore CPU assignments when a process is started by a NetBatch SCHEDULER process. As a result, jobs will be distributed over the preferred processors and will be under the exclusive control of the NetBatch SCHEDULER.

As a Tandem system grows and additional processors are added, the system manager need only change the SCHEDULER configuration rather than the CPU designations in every RUN command in every IN file.

Figure 3.
Relationship between JOBCLASSES, EXECUTORS, and CPUs.

Control of the Impact of Batch Processing

Knowing the history of all the applications running on the system, the Tandem system manager can control the impact of batch processing on OLTP. The objective is to set a priority ceiling level on batch processing to reduce any loss of performance in OLTP without sacrificing batch performance. The system manager must find that balance between maximum throughput of batch jobs versus the impact upon OLTP.

The DEFAULT-PRI parameter specifies the priority of the EXECUTOR program when it is running jobs whose PRI attributes are not specified at submit time. The default DEFAULT-PRI value is 120.

The PRI attribute specifies the run priority of the EXECUTOR program. Again, if not specified, it assumes the SCHEDULER DEFAULT-PRI value. The PRI value is best left undeclared so as not to exceed the priority level of the EXECUTOR program and that of the SCHEDULER.

Job Scheduling and Submission

NetBatch provides a facility that automates scheduling of batch jobs and maintains the logical dependencies between jobs. This feature allows the operations staff to concentrate on monitoring and controlling the batch environment and eliminate many of the operator-dependent functions. Once jobs are submitted to the SCHEDULER process for execution, the scheduling criteria are defined by the parameters described below.

Immediate Execution. The AT attribute dictates the specified date and time of execution even if a temporary EXECUTOR must be created. AT demands the job be executed exactly when scheduled. If the assigned EXECUTOR(s) is not available, the SCHEDULER creates a temporary EXECUTOR for the execution of this job as long as the DEFAULT-EXECUTOR-ELEMENTS are not exceeded.

The system manager may use this attribute for jobs that are time-critical, for example, a job that performs a shutdown of a bank's ATM network at 2:00 a.m. to ensure that collected transactions are posted during a given processing window.

Time Dependencies. The AFTER attribute indicates a job will run after a specific time and/or date. Most jobs will fall under this category. As long as an EXECUTOR is available and all scheduling criteria are met, this job will execute.

The EVERY attribute dictates a job executes every specified number of days or hours:minutes. For example, a job that runs every three days would be set with this attribute.

The WAIT attribute dictates a job wait a specified time period before execution.

The CALENDAR attribute specifies a calendar file that contains dates and times when the job is to be run. This function allows the system manager to submit jobs with complex scheduling requirements, for example, a job that runs every day at 6:00 a.m., except weekends and holidays.

Job Dependencies. The WAITON attribute specifies that a dependent job will not execute until released by (up to eight) master jobs. This feature requires TACL or BPROC as the EXECUTOR program of the master job.

WAITON offers the system manager two important functions. First, it ensures that a dependent job will not run prior to the master jobs completing; for example, a job that backs up a database to tape will not be initiated until end-of-day processing is completed.

Secondly, it ensures that dependent and subsequent jobs do not run if an error occurs with the master job(s). Without this feature, an error may cause a file to be corrupted but jobs will continue to execute until the entire day's run is complete, only to find it must be rerun.

On-Hold Job Submission. The HOLD attribute specifies that a job is held rather than executing immediately. This requires the operator to manually release the job prior to executing. An operator may submit all end-of-day processing jobs on hold in order to review them prior to execution.

Override Job Scheduling Option. After submission, the system manager may promote a job for early execution through the RUNNOW and RUNNEXT commands. Available to SUPER ID users, these commands override the SELPRI (Selection Priority) attribute of a job but not the scheduling attributes AFTER, AT, EVERY, or CALENDAR.

The RUNNOW command executes a job immediately even if a temporary EXECUTOR must be created. The RUNNEXT command executes a job immediately after the currently executing job has ended.

Batch Processing Visibility and Control

NetBatch provides the systems manager and operator (through the use of any terminal in the network) with a facility to monitor and control all batch processing on a distributed Tandem environment. To control the processes it initiates, NetBatch takes advantage of several GUARDIAN 90, version C00, enhancements. Among these are GMOM ("godmother," or ancestor monitoring process), completion codes, and the SPOOLER. The batch processing visibility and control features include monitoring, GMOM processing and handling, error-trapping, and SPOOLER management.

Monitoring. NetBatch-Plus gives the operator information on all the jobs executing in the Tandem batch environment. One group of commands allows the operator to see the status of the BATCHCOM objects:

- *STATUS SCHEDULER* displays configured and active attributes for EXECUTORs, jobs, JOBCLASSes, total processes, and tape drives. (See Figure 4a.)

Figure 4

```

BATCHCOM - T9190C10 - (15Mar88 ^ 11Apr88)
COPYRIGHT TANDEM COMPUTERS INCORPORATED 1986, 1987, 1988
SERVER VERSION: C10
BATCH SERVER - T9190C10 - 15MAR88 - C10 VERSION

(a) > STATUS SCHEDULER
    SCHEDULER STATUS for $ZBAT

    SCHEDULER LOGFILE: $SYSTEM.SYSTEM.logaad

    EXECUTOR      JOB      JOBCLASS      PROCTBL
    MAX 500      MAX 2000      MAX 500      MAX 5000
    CONFIG 25      CONFIG 500      CONFIG 100      CONFIG 2000
    FREE 22      FREE 499      FREE 98      FREE 2000
    OFF 0      READY 0      OFF 0      ACTIVE 5
    ON 3      EXECUTING 1      ON 3      SUSPENDED 0
    ACTIVE 1      SPECIAL 0
    STOP 0      TIME 1
    DOWN 0      EVENT 0
    DELETE 0      SUSPENDED 0
                   RUNNEXT 0
                   RUNNOW 0
                   TAPE 1

    2 TAPEDRIVES CONFIGURED; 1 IN USE.

(b) EXECUTOR STATUS

    EXECUTOR      CPU  STATE  JOB
    EX-1          0  ACTIVE  47
    EX-2          1  ON
    EX-3          2  ON

(c) JOB STATUS

    JOB  JOBNAME  SEL  USERID  CLASSNAME  STATE
    47  JOB1      3    64, 41  DEFAULT    EXECUTING
  
```

- *STATUS EXECUTOR* displays the state of each EXECUTOR, CPU designation, and total number of jobs executing. (See Figure 4b.)
- *STATUS JOB* displays, for each job name, the state of each job, queue job number, selection priority, owner ID, and job class designation. (See Figure 4c.)

Figure 4. Commands that show the status of BATCHCOM objects. (a) *STATUS SCHEDULER* display of schedule configurations. (b) *STATUS EXECUTOR* display of EXECUTOR activity. (c) *STATUS JOB* display of job activity.

Figure 5

```
JOB STATUS

BATCHCOM - T9190C10 - (15Mar88 ^ 11Apr88)
COPYRIGHT TANDEM COMPUTERS INCORPORATED 1986, 1987, 1988
SERVER VERSION: C10
BATCH SERVER - T9190C10 - 15MAR88 - C10 VERSION

> INFO SCHEDULER
SCHEDULER ATTRIBUTES $ZBAT
      backupcpu: *
      at-allowed: on
      submit-allowed: on
      executor-program: $DATA.TACL10.TACL
      default-pri: 120
      default-selpri: 3
      default-maxprintlines: 50000
      default-maxprintpages: 500
      tapedrives: 2
      default-out: $S.#BATCH.NBJOB
      default-class: DEFAULT
default-executor-elements: 25
default-job-elements: 200
default-jobclass-elements: 50
default-proctbl-elements: 500
```

Figure 5.
INFO SCHEDULER
display of static information of the schedule configuration.

Another group of commands allows the operator to see the configuration of the BATCHCOM objects:

- *INFO SCHEDULER* displays the configured attributes of the SCHEDULER. (See Figure 5.)
- *INFO EXECUTOR* displays two active attributes, CLASS and CPU.
- *INFO JOBCLASS* displays the active attribute INITIATION.
- *INFO JOB* displays the active attributes set at job submission or as default attributes of the associated job class.

GMOM Process-Handling. Among the enhancements of GUARDIAN 90, version C00, was the introduction of the GMOM-CRTPID-JOBID (Godmother-Creation Time Process ID-Job ID) parameter in the NEWPROCESS call. This parameter gives the NetBatch SCHEDULER the ability to track and control all processes that belong to a

given job. This translates to the SCHEDULER being able to SUSPEND or STOP all processes of an executing job(s), displaying ACTIVE and SUSPENDED processes under STATUS SCHEDULER PROCTBL, or receiving creation and completion messages of all processes with the same GMOM-CRTPID-JOBID.

In Figure 6, the SCHEDULER is a GMOM process. When the SCHEDULER starts a job, it does a NEWPROCESS call. The SCHEDULER passes the EXECUTOR program TACL name as the filename and a non-zero value for the GMOM-CRTPID-JOBID parameter of the NEWPROCESS call. This value resides in the PCBX (process control block extension) and is expressed as the ancestor's (creator) or the GMOM-CRTPID, and a non-zero value. The GMOM-CRTPID of 1,100 and the JOBID value 1 is passed to the TACL and to the subsequent ENFORM™ and QP (query processes).

The EXECUTOR program in turn initiates other processes using implicit or explicit calls to NEWPROCESS. The relationship between these processes and the SCHEDULER is determined by the value of the GMOM-CRTPID-JOBID.

When the process is created, it sends a start message to the immediate ancestor and GMOM, which contains the GMOM-CRTPID-JOBID parameter as assigned by the SCHEDULER. The newly created process is now linked to the ancestor and GMOM. The GMOM can therefore keep track of all new processes that belong to this job. When a process is terminated, STOP/ABEND messages are sent to the GMOM as well as the ancestor of the process. This internal GUARDIAN feature gives a NetBatch SCHEDULER complete tracking and control over the jobs and processes it initiates.

To dissociate a process from the SCHEDULER requires that the RUN command in the IN file specify the GMOM-CRTPID-JOBID value be 0. In Figure 7, the value of the PATHWAY PATHMON process is 0, and the PATHMON that is started is dissociated. The SCHEDULER has no knowledge of this process. This feature is useful in instances where a process is started by NetBatch but must not be controlled by it, for example, a bank initiates a batch job that activates a Purchasing PATHWAY application.

Error-Trapping. There is a facility that the system manager can incorporate into an IN file to automatically trap and recover from processing errors. Every job process that executes returns a process completion code (a value that indicates how well the process ran) to its ancestor process and the SCHEDULER whenever a STOP/ABEND call is made. The system manager can code a routine to test the value of the completion code so the job can be routed to a conditional course of action.

For example, if TACL is the EXECUTOR program of the job, the #IF..THEN...ELSE or #CASE statements can test for a particular value of the completion code, use #OUTPUT to display a message back to the TACL EXECUTOR program output file or RUN another program to invoke error recovery procedures. This action may be a simple restart or recovery routine.

Error-trapping not only reduces the dependence on operators to resolve simple problems, but also ensures that subsequent action is taken in time and not impact the throughput of jobs.

SPOOLER Management. Besides the enhancements made to C00 GUARDIAN, the SPOOLER was also modified to provide new management features for batch jobs. A new SPOOLER job definition, SPOOLER BATCH JOB, consists of a user log, the EXECUTOR program OUT file (discussed in the next section under Audit Trails), and all job-process output from executing a single NetBatch job.

By default, the maximum number of concurrent SPOOLER BATCH JOBS is 256. This limit is set by the SPOOLER configuration. Should this limit be exceeded, the oldest SPOOLER BATCH JOB will be dropped and replaced by the next job.

The EXECUTOR program initial process sets four attributes of every SPOOLER job:

- *GMOM-CRTPID-JOBID* (described earlier).
- *FORM*, specifying a particular printing device or paper.
- *OWNER*, indicating the user who made the request to the SPOOLER.
- *LOC*, specifying the logical destination of a SPOOLER job.

Figure 6

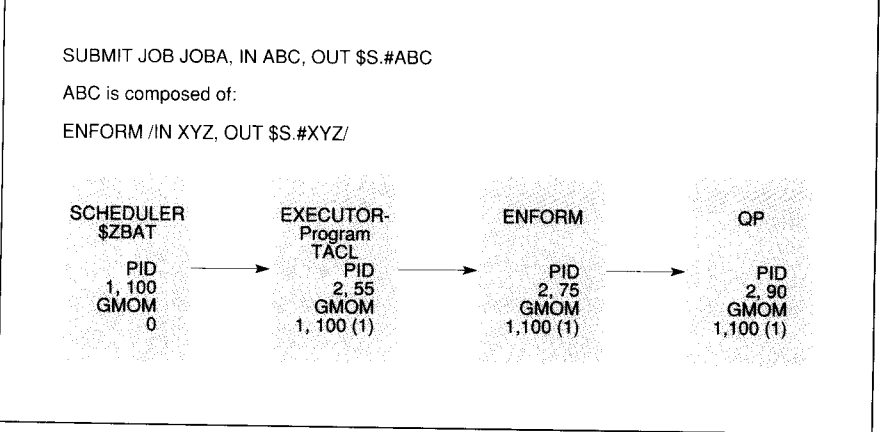


Figure 7

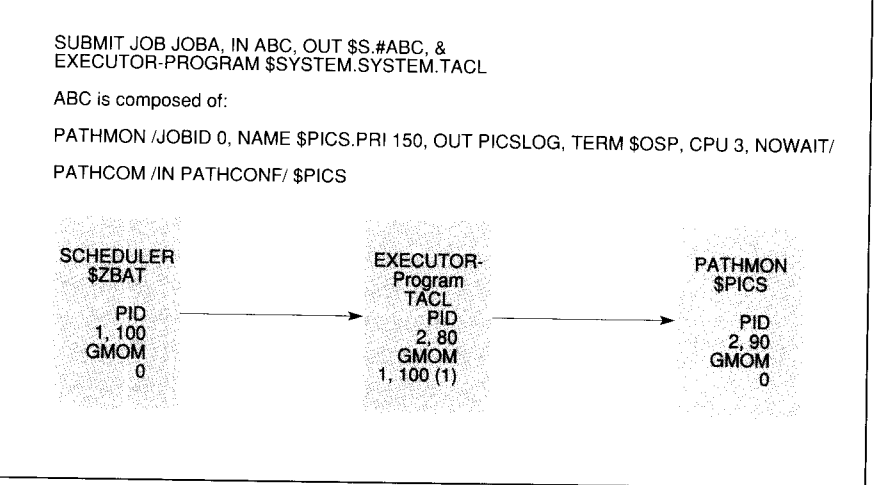


Figure 6.

JOBA query process ties back to the ancestor process through the GMOM process handling.

Figure 7.

Specifying JOBID 0 in the PATHMON command dissociates the GMOM process handling.

Figure 8

```
> SUBMIT JOB JOBA, IN UPDATE, OUT $$.#POSTING

JOB CONTROL FILE = UPDATE

COMMENT JOBA --- RUN THE ACCOUNT MASTER UPDATE PROGRAM

ASSIGN TEMP-BILLING-FILE, $DATA.TEMP.BILLING
ASSIGN ACCTMSTR          , $DATA.ACCTDATA.ACCTMSTR

ADD DEFINE =OUT ^ SPECIAL, CLASS SPOOL, LOC $$.#LP1, REPORT INVOICE
ADD DEFINE =OUT ^ AR, CLASS SPOOL, LOC $$.#LP2, REPORT "AR REPORT"
ADD DEFINE =OUT ^ AP, CLASS SPOOL, LOC $$.#LP2, HOLD ON, COPIES 3

RUN $DATA1.OBJ.ACCTUPD

PURGE $DATA.TEMP.BILLING
```

Figure 8.
JOBA job control file contents.

The SPOOLER uses these attributes of the output to assign the BATCHID. Within a single batch job, multiple job-process output that have the same attributes are assigned the same BATCHID. Those with dissimilar attributes are assigned another BATCHID.

Consider the example in Figure 8 where JOBA is submitted to be executed through BATCHCOM. There are three process outputs from the execution of this billing program. The invoices require line printer #LP1, which is stocked with continuous invoice forms, and the accounts receivable and accounts payable reports require line printer #LP2, which is stocked with wide stock paper. DEFINES were used in this example to modify the key attribute LOC to highlight the assigning of different BATCHIDS for different process outputs.

As shown in Figure 9, reports that require the same form are grouped with the same BATCHID. This allows an operator to output these reports to a single printer location with minimum effort. Note that both the PERUSE and SPOOLCOM interfaces have a column for the BATCHID numbers. Also, SPOOLER JOB 100 is the User Log and JOB 101 is the EXECUTOR program OUT file.

Two batch-oriented SPOOLER commands available to PERUSE and SPOOLCOM interfaces are LINK and UNLINK. These commands allow an operator to add or delete a BATCHID to SPOOLER jobs. The LINK command links a current SPOOLER job to a current SPOOLER batch job if the key attributes of the SPOOLER job match those of the SPOOLER batch job. The UNLINK command dissociates a SPOOLER job from the current SPOOLER batch job.

Audit Trail of Batch Activities

NetBatch provides two detailed and integrated logs of activities within the batch processing environment. As jobs are processed through the EXECUTOR program, the SCHEDULER records all major events that occur during the submission and execution of all jobs.

The User Log file is an unstructured disk file created when the SCHEDULER process is activated (cold start). It holds a special file code 847 reserved for files created with NetBatch. Figure 10 is the User Log file, LOGAAD record of job ALOHA1.

As jobs are processed through the EXECUTOR program, the SCHEDULER records all major events that occur during the submission and execution of each job. The User Log is usually a SPOOLER file and contains the following:

- Job submit, start, and stop time.
- Process start and stop time.
- Process CPU time.
- Job-accumulated process CPU time.
- Messages to and from the operator.
- Job restart notification.

At the same time, the EXECUTOR program logs its events of the processing involved to the SPOOLER as the EXECUTOR program OUT file. These logs provide a facility to track batch processing errors and allow the operator to react quickly and effectively in initiating the proper recovery procedures. With a central source of information, recovery time is not wasted.

Figure 9

```

PERUSE - T9101C00 - (15JUL87)
                                SYSTEM \SUPPORT
JOB  BATCH  STATE  PAGES  COPIES  PRI HOLD  LOCATION  REPORT
100  42     READY   1       1       3       #POSTING  GL      MGR
101  42     READY   1       1       3       #POSTING  GL      MGR
102  43     READY  90       1       4       #LP1      INVOICE
103  44     READY  30       1       4       #LP2      AR REPORT
104  44     HOLD   35       3       4       #LP2      GL      MGR
_EXIT

SPOOLCOM - T9101C00 - (15JUL87)
)JOB (OWNER 64,255)
                                SYSTEM \SUPPORT
JOB  BATCH  STA  FLAGS  OWNER  TIME  COPY  PAGE  REPORT  LOCATION
100  42     RDY  3     64,255  19:40  1     1     GL      MGR  #POSTING
101  42     RDY  3     64,255  19:40  1     1     GL      MGR  #POSTING
102  43     RDY  4     64,255  20:45  1     90    INVOICE #LP1
103  44     RDY  4     64,255  21:00  1     30    AR REPORT #LP2
104  44     HLD  4     64,255  21:05  3     35    GL      MGR  #LP2
)EXIT

```

Figure 9.

PERUSE and SPOOLCOM displays of the NetBatch job.

Conclusion

With an understanding of the features and benefits of the NetBatch product, a user can improve the impact of batch processing so it is run effectively and efficiently on Tandem's multiprocessor architecture. Using NetBatch as a foundation for batch processing, a user is able to balance the batch processing load, control the impact on OLTP, utilize the job scheduling and submission functions, control and monitor jobs before and during execution, and have an audit trail of all activities of the processing. The features and benefits of NetBatch complement Tandem's strength in OLTP. Together, these products make Tandem systems a total solution for any company's data processing needs.

References

NetBatch User's Guide. Tandem Computers Incorporated. Part no. 18097.

NetBatch Independent Study Program Guide, Beta version. Tandem Computers Incorporated.

Acknowledgments

I would like to thank the members of our Tandem community for contributing material and ideas on NetBatch and for reviewing this article. Special recognition to David Cooper and Vickie Rose for the diagrams and examples.

Dean K. Kaumaakani Wakashige is a senior staff analyst for the Customer Support Organization. Prior to joining Tandem in 1987, Dean was the support manager for a Tandem third-party vendor and a systems analyst for a major bank.

Figure 10

```

NetBatch SCHEDULER/MONITOR - T9190C10 - (15Mar88 ^ 11Apr88)

880810 13:44:40>Opened by LSS.DEAN from program $SYSTEM.SYSTEM.BATCHCOM
880810 13:45:17>JOB 12 being placed on list
880810 13:45:17>JOB 12 placed on list ready
880810 13:45:18>ADDING JOB ALOHA1
880810 13:45:21>ADDING $Z047 4,154 TO PROCTBL FOR JOB 12
880810 13:45:22>EXEC STARTED for LSS.DEAN ALOHA1 $Z047 4,154
880810 13:45:48>\133 110 6372 7,152 started on behalf of job 12
880810 13:45:48>ADDING \133 110 6372 7,152 TO PROCTBL FOR JOB 12
880810 13:45:50>ABEND received from \133 110 6372 7,152
880810 13:45:51>DELETING \133 110 6372 7,152 FROM PROCTBL FOR JOB 12
880810 13:45:52>\133 110 7161 6,179 started on behalf of job 12
880810 13:45:52>ADDING \133 110 7161 6,179 TO PROCTBL FOR JOB 12
880810 13:46:56>Closed by LSS.DEAN
880810 13:47:53>STOP received from \133 110 7161 6,179 COMPLETION CODE 0
880810 13:47:55>DELETING \133 110 7161 6,179 FROM PROCTBL FOR JOB 12
880810 13:48:03>STOP received from $Z047 COMPLETION CODE 0
880810 13:48:05>EXECUTOR-PROGRAM has stopped for JOB 12 ALOHA1
880810 13:48:08>DELETING $Z047 4,154 FROM PROCTBL FOR JOB 12
880810 13:48:10>FLUSH ^POOL ^BUFFERS for job 12
880810 13:48:11>DELETING JOB ALOHA1

```

Figure 10.

User log file output for job ALOHA1.

Tandem 5200 Optical Storage Facility: A Hardware Perspective

Traditional methods of archiving information make use of magnetic tape, microfiche, and paper as the storage media.

The 5200 Optical Storage Facility (5200 OSF) brings new technology to the storage and retrieval of archived information. It greatly reduces the amount of floor space needed to store archives and allows on-line random access with improved data integrity.

This article describes the hardware components of the 5200 OSF and how they interact, as well as how the performance of individual components benefits the storage and access of archived information. Refer to an earlier article by Lauryl Sabaroff in the *Tandem Systems Review* (February 1988) for a description of the basic concepts of optical storage techniques.

5200 OSF Hardware

The 5200 OSF hardware comprises optical cartridges, disk drives, a disk changer, electronic assemblies, and an I/O controller. A logical diagram of the components is shown in Figure 1.

Cartridges

Cartridges are stored in a 32-slot carousel. Each cartridge is removable and contains a single platter of media. Data is stored on both sides of the platter in a single spiral groove that forms a track at each complete revolution. (See Figure 2.) A cartridge has 41,300 tracks per side; each track contains 64 sectors, two of which are spares. Each sector comprises a header, user data, flags, and error correction code.

Disk Drives

The 5200 OSF has two optical disk drives. Although optical and magnetic disk drives are similar, there are several fundamental differences. Optical disk drives contain a laser with optics for a read/write head and are designed to work with removable media. While the seeking mechanism of a magnetic drive relies on a dedicated servo surface, optical drives make use of the spiral groove on the cartridge.

Two drives allow overlapped operations. The drives do *not* form a mirrored volume. However, when one drive fails, data can be accessed by the other drive.

Figure 1

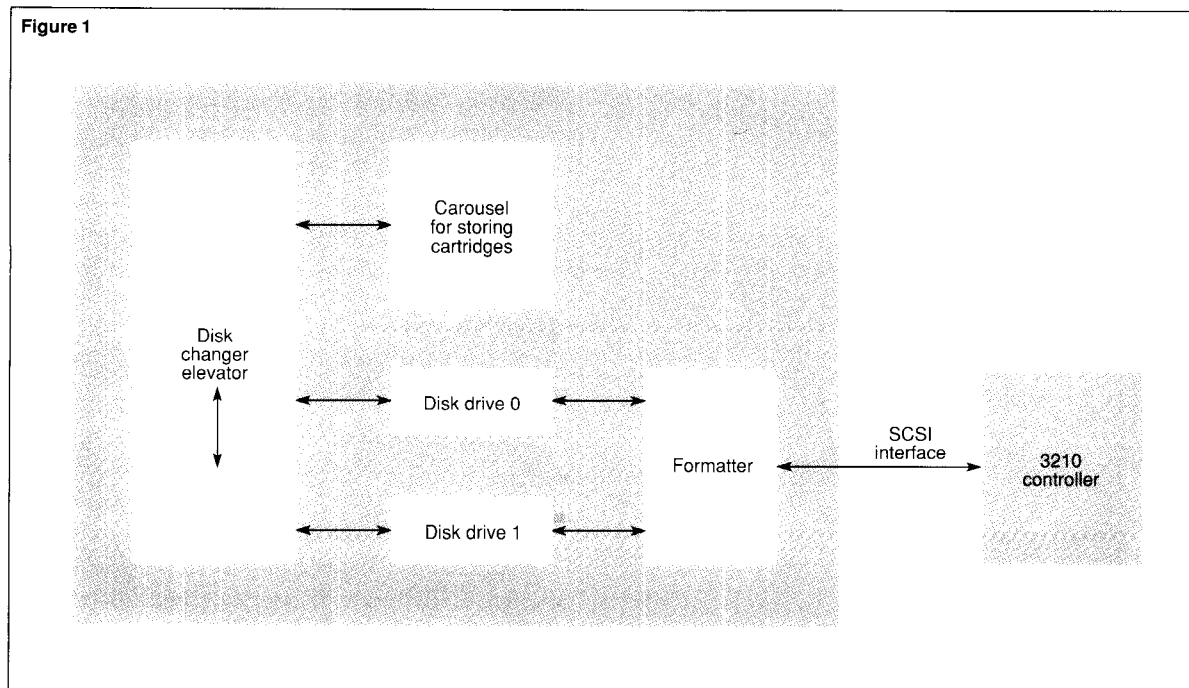


Figure 1.
5200 OSF logical
diagram.

Disk Changer

The disk changer comprises the carousel and an elevator that loads cartridges from the carousel to a disk drive and vice versa. The disk changer also makes it easier for an operator to add or remove cartridges from the carousel.

Two PUP (peripheral utility program) commands, INSERT and EJECT, are used to add or remove cartridges from the 5200 OSF. The EJECT command allows cartridges to be stored off-line. Cartridges stored off-line can be re-inserted into any 5200 OSF.

Controller

The 5200 OSF communicates with the system via a Tandem 3210 controller which, like the 3207 controller, employs lock-stepped microprocessors and self-checking logic. The 3210 controller uses the differential version of the small computer systems interface (SCSI), allowing the 5200 OSF to be placed up to 75 feet from the CPU cabinet.

The 3210 controller requires two addresses—one for each disk drive. Addresses are *not* allocated on a cartridge basis.

To the optical disk process, each side of a cartridge is a volume, for a maximum of 64 volumes. The process assigns 64 logical device numbers to the 5200 OSF; however, only two physical addresses are required—one for each disk drive.

Storage Capacity

The 5200 OSF has a formatted storage capacity of 84 Gbytes, stored in a footprint of only 9.7 square feet.

The high storage density begins with the bits on the cartridges that pack bits at 19,000 per inch and tracks at 16,000 per inch. The resulting areal density is 304 Mbytes per square inch (psi). The storage density is further increased by storing data on both sides of a cartridge.

Figure 2

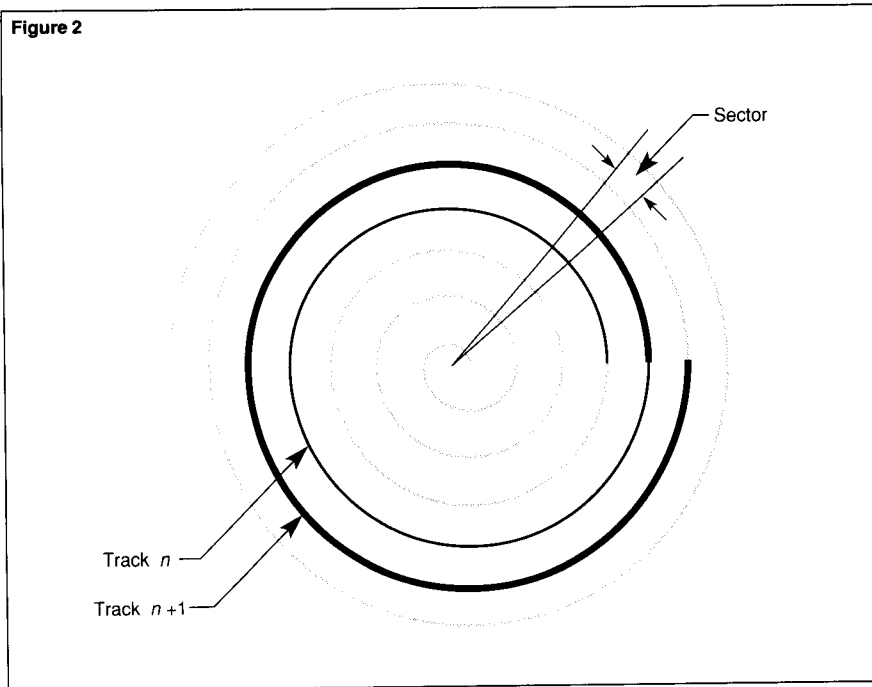


Figure 2.

The platter. A single spiral groove forms a track at each revolution.

Unlike magnetic drives, optical drives are tolerant of dust particles on the media. Optical heads keep a 0.04-inch gap between the head and media, about 2500 times more than the gap between a magnetic disk head and media. As a result, optical heads do not “crash” and media does not need to be sealed in the same assembly as the head. Therefore, optical media is removable.

Thirty-two cartridges can share two disk drives, resulting in increased storage density with a lower cost increase. Cartridges can be stored outside the 5200 OSF in an office environment. When stacked outside of the 5200 OSF, the cubic storage density is about 26 Gbytes per cubic foot, compared to 1 Gbyte per cubic foot for GCR¹ tape.

¹Group Code Recording.

On-line Random Access

On-line random access is a new feature in archiving. Any cartridge can be picked by the disk changer and loaded into any disk drive. The disk drive can then seek directly to the desired track and access data. Both the disk changer and the disk drive seek mechanism contribute to random access.

Disk Changer

There are three types of disk changer operations relating to the cartridge:

- Load and unload.
- Flip.
- INSERT and EJECT.

A load moves a cartridge from the carousel to a drive. The elevator moves up or down as required to pick a cartridge from the carousel and load it into a disk drive. After a cartridge is loaded, the drive spins up the cartridge and seeks to track 0. An unload moves a cartridge from a drive to the carousel.

The elevator also flips cartridges so that either side can be accessed. Cartridges can be flipped while the elevator is moving up or down.

Using PUP commands, the carousel can be rotated 90 degrees for access either by the elevator or a user. When the carousel faces the elevator, user access is prevented by an operator access door. To insert or eject a cartridge, the carousel must be rotated to face the operator access door, releasing the door interlocks.

A PUP INSERT command is issued to add cartridges. When the door is open, the user can place cartridges in empty slots. Sensors in each slot tell the disk process about the location of the cartridges.

The PUP EJECT command is directed toward a specific cartridge, which is identified by a yellow LED when the operator access door is opened. If the wrong cartridge is removed, sensors alert the optical disk driver, and the disk driver notifies the user. The driver also verifies the volume label when the cartridge is loaded in a disk drive.

Only one of the three disk change operations can be performed at a time.

Seek

Both optical disk drives and magnetic drives can perform random seeks, but the methods used are different. Magnetic disks have a dedicated surface called the "servo platter" to help seek and follow a track as the disk spins around. Optical media uses grooves to assist in seeking and track following. The groove diffracts the laser beam, and the head uses these light patterns to stay on track.

There are two seeking modes: *coarse* and *fine*. To perform a coarse seek, the drive estimates the distance and moves the head near to the target track. The formatter then reads the header of the track and determines which track it actually reached.

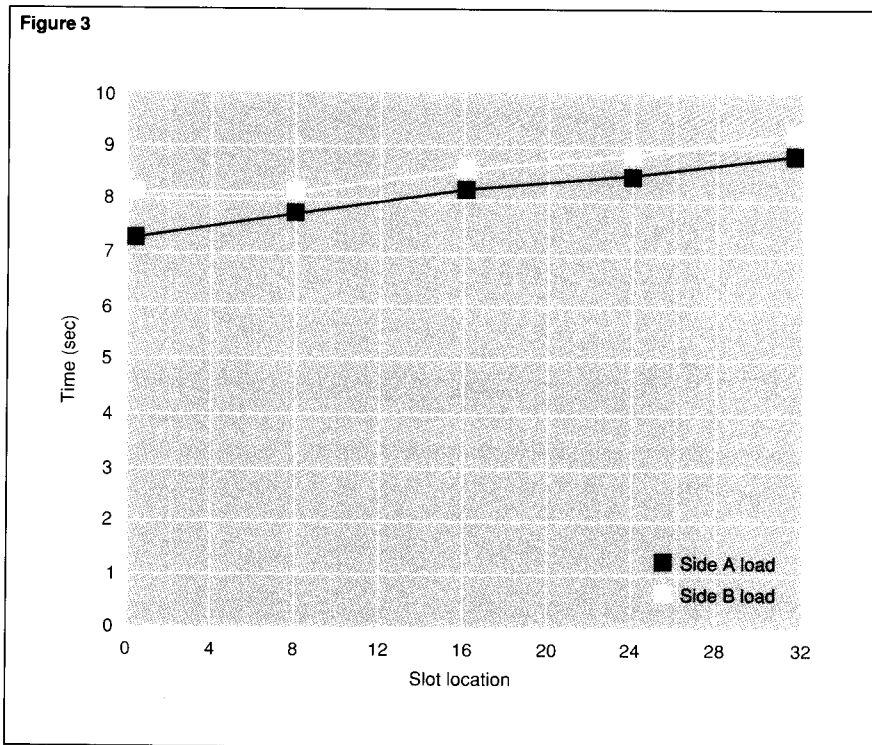
The head is then moved exactly to the right track by performing a fine seek. A fine seek uses a rotating mirror in the head. As the mirror is deflected, the laser beam crosses tracks one by one. The diffracted pattern from the grooves goes alternately dark and bright for each track. The drive counts the number of tracks crossed and stops once the beam is on the desired track. The head assembly is then moved to position the head directly over the desired track. The formatter confirms the head position by reading header fields on the track.

Performance

Archived data is typically stored off-line; data retrieval time can range from minutes to days. By automating the storage and retrieval process, the 5200 OSF improves the speed of cartridge and data access.

Cartridge Access Time

Cartridge access operations include loading, unloading, and flipping a cartridge. Typically, a cartridge is unloaded from a drive and another cartridge is loaded in the same drive in a process known as "cartridge exchange." The time required to perform cartridge access is a function of cartridge location. (See Figure 3.)



Minimum, average, and maximum times needed to perform a cartridge access operation are shown in Table 1. Much of this time is consumed by the disk drive. The drive takes 4 seconds to spin a cartridge up and 3.7 seconds to spin a cartridge down.

Figure 3.
Cartridge load time as a function of location.

Table 1.
Minimum, average, and maximum times required to perform a cartridge access operation.

Operation	Time (sec)		
	Minimum	Average	Maximum
Load	7.0	8.4	10.0
Unload	7.0	8.4	10.6
Exchange	14.0	16.8	20.6
Flip	11.4	12.8	14.2

Figure 4

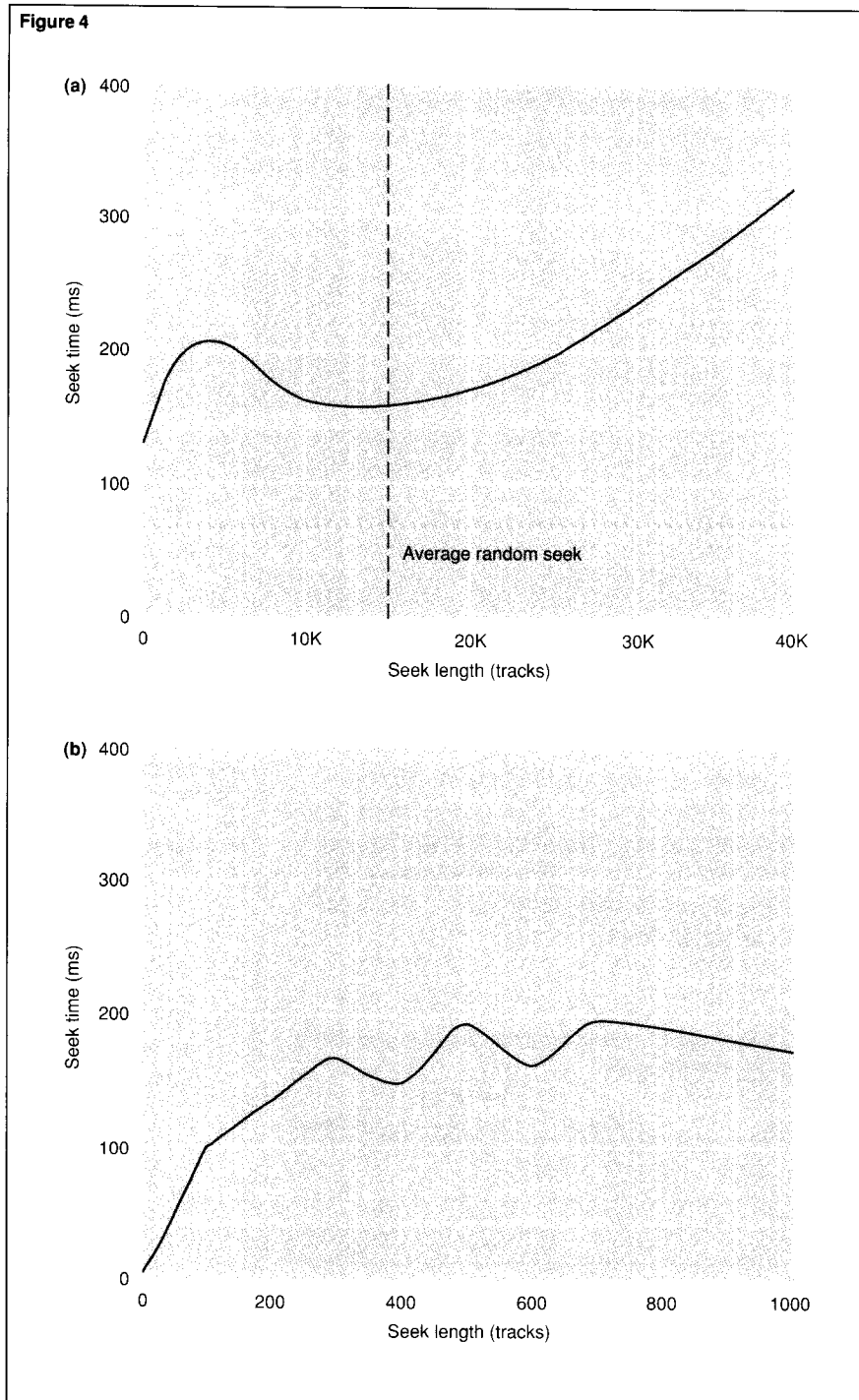


Figure 4. Seek performances profiles. (a) Macro seek times. (b) Micro seek times.

Seek Time

The tracks are segments of a large spiral, so moving from track to track does not require a seek. Therefore, large amounts of data can be transferred without incurring seek delays.

Figure 4a shows the seek performance profile for seeks ranging from 0 to 40,000 tracks. The maximum seek is 41,300 tracks and takes 330 ms. The mean seek time is 190 ms.

In Figure 4b, the scale of the seek profile for seeks ranging from 0 to 1000 tracks is expanded. Fine seeks are performed when the seek length is less than 100 tracks. The actuator is not moved for fine seeks—a mirror in the head assembly is deflected. Therefore, the time to seek less than 100 tracks is only about 1 ms per track.

Latency

Latency is determined from the rotational speed of the disk and equals the amount of time required to access the desired sector once the drive seeks to the designated track.

Optical cartridges are spun at 600 rpm, or one rotation every 100 ms. For writes, as expected, the average latency is 50 ms, or half the maximum. Sectors are recorded in the order they are transmitted from the controller.

However, during reads, the formatter starts reading as soon as it encounters any of the sectors requested by the user. This is supported by the organization of data buffers within the formatter. The formatter has two buffers (A and B), as shown in Figure 5. Each buffer can store one track of information. At any given time, a buffer contains data from only one track. Within each buffer, there are specifically designated areas for each sector.

As soon as it encounters any of the sectors requested, the formatter begins storing them in their designated areas. It then transmits them to the controller in the order requested.

Therefore, when reading, the latency experienced depends both on the length of the transfer and on whether the data is present on a single track. If “ n ” sectors must be transferred from a track, the average latency to access data is equal to $50 \times ((65 - n) / 64)$ ms.

Figure 6 shows how 30 Kbytes of data are read. The required 30 Kbytes are stored in 60 sectors, from sector 2 through sector 61. The gray line shows when the data requested is being stored in a formatter buffer. The average latency to read 30 Kbytes from a track is 4 ms.

When the requested data is stored over two tracks, latency is incurred for each track. Figure 7 shows how the same amount of data is spread over two tracks. The dashed line indicates periods when data not requested is being skipped. Data from the second track is not read until all required sectors from the first one have been read. The average latency when reading data spread over two or more tracks is 50 ms.

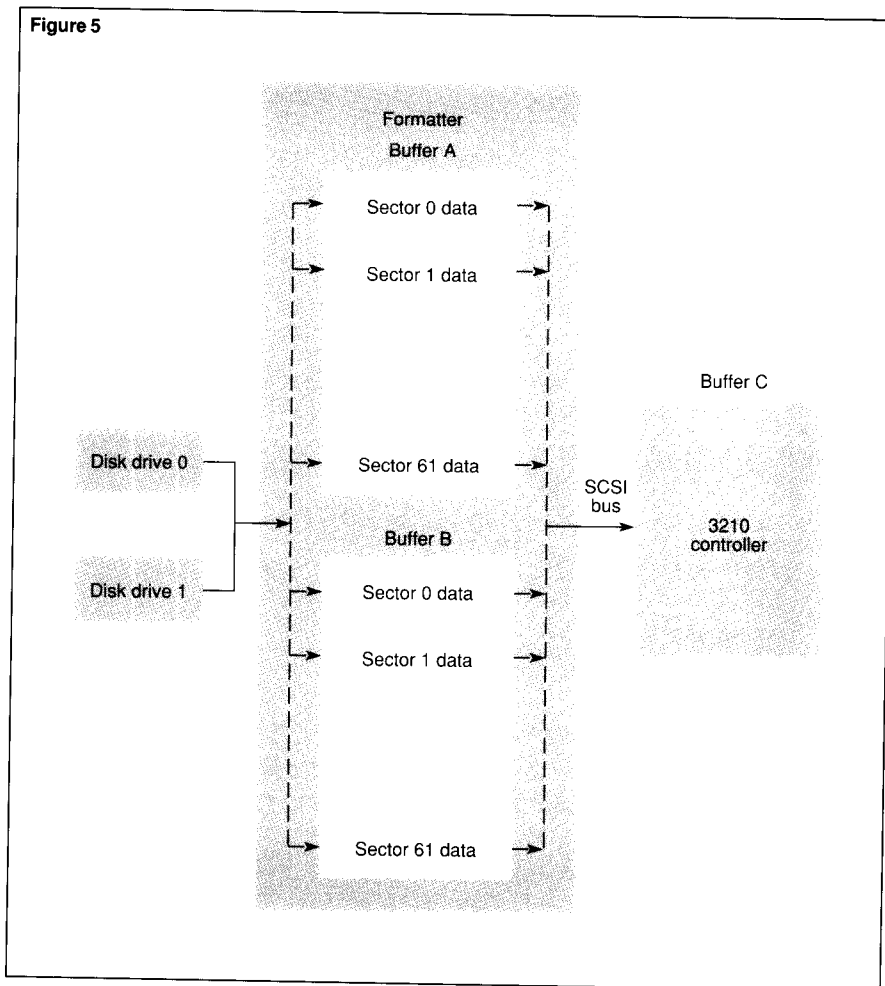
Data Transfer Rate

The optical disk drive transfers data to and from the media at 3.5 Mbits per second, or 442 Kbytes per second. This is the actual rate at which data is transferred, including user data and header, flag, and error correction code (ECC) fields. User data is transferred at an average rate of 317 Kbytes per second, i.e., each track contains 31,744 bytes of user data and the disk rotates one full track every 100 ms.

During write operations, data is written during one revolution and verified on the next revolution; the average data transfer rate when writing is 159 Kbytes per second. When reading, data can be read on every revolution. Therefore, the read transfer rate is 317 Kbytes per second.

Long Data Transfer

To sustain the 317-Kbyte read transfer rate, data must be read on every revolution. The spiral arrangement of tracks allows the disk drive to read data on every revolution without seeking—it simply follows the spiral. For long data transfers, hardware, firmware, and the disk process are designed to take advantage of this capability.



Long data transfers (called serial reads) use buffers A and B in the formatter and buffer C in the controller. (Refer to Figure 5.) Buffers A and B can operate independently. While the formatter is reading data from a drive to a buffer, the other buffer can send data to the controller.

Figure 5.
Buffering scheme in the optical disk formatter.

Figure 6

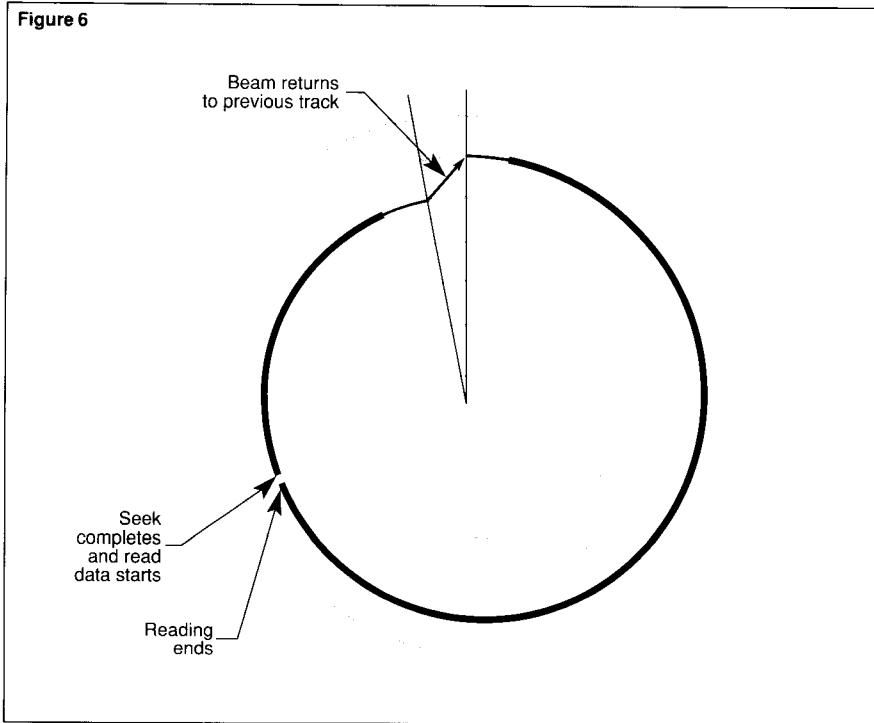


Figure 6.
Full track reads have no latency.

First, the disk process informs the controller that a serial read is about to begin. The controller asks the formatter to read a very large amount of data (up to 4 Mbytes). The formatter reads data into buffer A. When a track boundary is crossed, the formatter reads the next track into buffer B. Simultaneously, buffer A is emptied into the controller buffer. When the next track boundary is crossed, buffer A is filled while buffer B is emptied into the controller buffer.

Meanwhile, the controller stores data from the formatter in its buffer, and transfers it to the CPU as data is requested. To avoid slipping revolutions, data is read in advance, before it is requested. This is called *read lookahead*. Data is transferred out to the CPU as read commands are received. The maximum amount of data permitted per read command is 30 Kbytes.

A revolution is slipped if neither buffer A nor buffer B is empty about 60 ms before the beginning of a new track. The mechanism works best when 30 Kbytes are transferred per read command. Also, a very busy system may not be able to read data in time to avoid slipped revolutions.

Overlapped Operations

The OSF allows overlapped operations for best performance. Data transfer on one disk drive can be overlapped with a disk change operation on the other drive. Read or write is permitted on one drive at a time. However, a disk change can occur while a read or write is in progress.

Read or write operations are allowed during an insert or eject operation. However, disk change operations are not permitted at this time.

Data Integrity

The 5200 OSF is designed to increase the integrity of archived information. The process of storing and retrieving data is more reliable because it is automated. Several techniques used by the 5200 OSF contribute to data integrity.

Error Correction Codes

Sixty bytes of ECC are stored with 512 bytes of user data. This allows the 5200 OSF to correct an error burst of up to 30 bytes. The error rate is one uncorrectable error in 10^{12} bits—equal to most magnetic disks.

To protect against undetected or miscorrected errors by the ECC, 2 bytes of cyclic redundancy check (CRC) are stored. Data transferred from the drive to the formatter is protected by both ECC and CRC. Data transferred from the formatter to the controller is parity protected. The next section shows how data is protected when writing.

Read after Write

Data is written to a sector only after reading the header flags and verifying that the sector has no data, i.e., is "blank," thus preventing data from being accidentally overwritten.

After user data is written, flags are inserted on the next revolution to indicate that the sector is not blank. The readability of the data is also checked at this time. Data is remapped to a spare sector, if necessary.

ECC verification of data after writing guarantees that the write was successful. The use of flags prevents accidental writes on written sectors and also prevents inadvertent reads of blank sectors.

Data Retention

Data stored on an optical disk can be read with an error rate of one uncorrectable error in 10^{12} bits for 10 years after it is written. Accelerated life tests have shown that the actual figure should be much higher—up to 30 years. Unlike other archive media, long-term cartridge storage does not require controlled environmental conditions.

Conclusion

Optical recording, robotics, and advanced electronics work together in the 5200 OSF to improve the techniques of storing and retrieving archived data. For the first time, entire archives can be kept on-line where they are accessible across a network and where they can be effectively managed under computer control. The 5200 OSF provides higher data integrity and better performance than competing archive storage systems.

References

Chandran, S. 1985. Introducing the 3207 Tape Controller. *Tandem Systems Review*. Vol. 1, No. 2. Part no. 83935.

Sabaroff, L. 1988. The Role of Optical Storage in Information Processing. *Tandem Systems Review*. Vol. 4, No. 1. Part no. 11078.

Acknowledgments

The author would like to thank Bruce Bailey and Steve Coleman for their advice on the article structure and technical content.

Anand Patel joined Tandem in 1983 as a disk development engineer. Prior to that, he worked for a major drum manufacturer. Anand graduated from Rensselaer Polytechnic Institute with an M.S. in Computer Engineering in 1981.

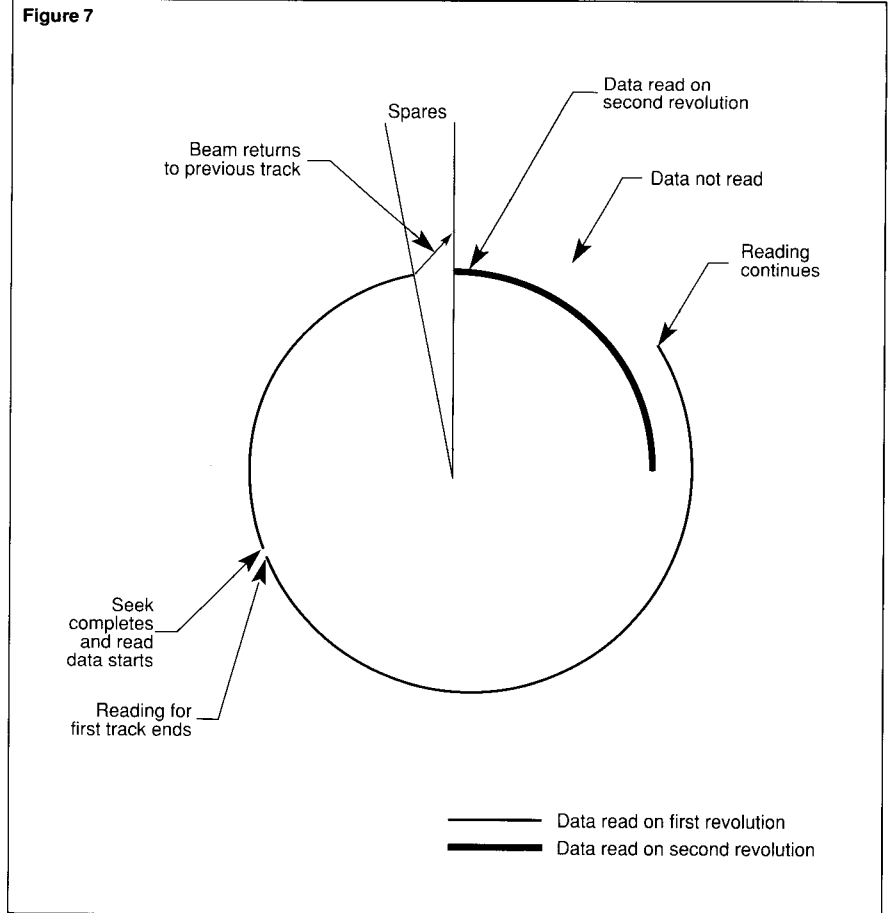


Figure 7.

Intertrack reads have latency.

Tandem 5200 Optical Storage Facility: Performance and Optimization Considerations

The 5200 Optical Storage Facility (5200 OSF) is a storage subsystem comprising two optical disk drives, 32 removable disk cartridges, and an automatic cartridge handler that moves cartridges back and forth from the cartridge storage rack to either of the two drives. The entire subsystem is enclosed in a compact cabinet known as a

jukebox. The preceding article, "Tandem 5200 Optical Storage Facility: A Hardware Perspective," by Anand Patel contains a more detailed description of the components of the 5200 OSF, as well as basic hardware performance information.

This article provides a performance analysis of the 5200 OSF that can be used in sizing and predicting the performance of an optical disk-based application.

Disk Drive Performance Issues

Of the variables discussed in the previous article, four have the most effect on the performance of the optical disk drive: data rate, latency, seek performance, and data buffering. The data rate of the drive was shown to be 317 Kbytes per second, which is sustainable on long transfers. Average latency on long transfers is 50 ms, but when transfers do not cross track boundaries, latency is reduced. Average length seeks were shown to be about 190 ms, and short seeks required about 1 ms per track. Data buffering allows overlapped transfers and reduces latency on short reads.

This section discusses the combined effects of these variables together with one additional parameter: aggregate transfer size.

Aggregate Transfer Size

Aggregate transfer size is the total number of bytes of data that are transferred as a result of a single read operation. This corresponds to the size of a file or an image record that is being retrieved; it is not related to the "block size" of data sent across the channel. Channel block sizes vary between 1 and 30,720 bytes, while aggregate transfers from the optical disk can vary between 512 bytes to over 33 Mbytes (64K sectors). When a read is initiated on an optical disk, the drive is told how many sectors are needed and the drive reads all the data at its maximum data rate (317 Kbytes per second) as long as it is not held off by the host during the transfer.

Figure 1 shows the effect of aggregate transfer size on drive performance using two different measures: net throughput and accesses per second. Net throughput in Kbytes per second is computed as follows:

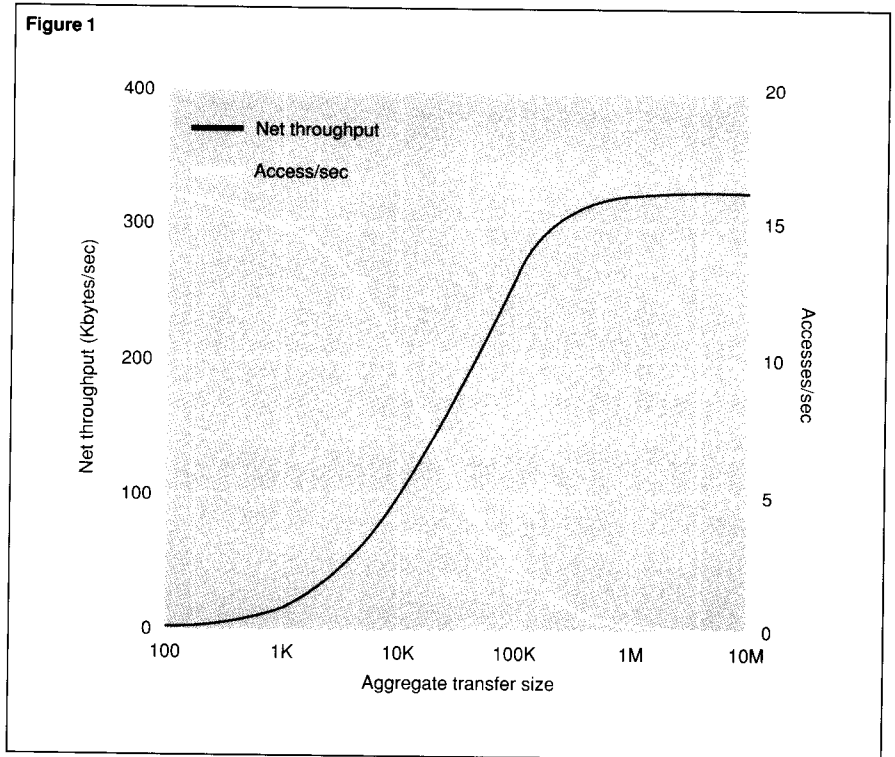
$$\text{Net_throughput} = \frac{\text{Aggregate transfer size}}{\text{Time}}$$

where aggregate transfer size is the average number of bytes in each read, and time is the number of seconds it takes to complete the transfer of data from the disk (including latency, seek times, and data transfer rates).

Accesses per second is computed as follows:

$$\text{Accesses/sec} = \frac{\text{Aggregate transfer size}}{\text{Net_throughput}}$$

These expressions are just inverse relationships, but give two very different ways of looking at the subsystem. When it is important to move a lot of data through the drive, large block sizes are absolutely necessary to make the most efficient use of the theoretical maximum data rate. As shown in Figure 1, 100-Kbyte transfer sizes are fairly optimal since this is the point where the throughput curve begins to roll off. Few throughput gains are made beyond 100-Kbyte transfer sizes.



For applications that require more frequent random access to data, smaller blocks appear to provide reasonable access frequency. However, for a more complete picture of accessibility, the combined effects of seek, latency, and data rate also need to be considered.

Figure 1.
Net throughput and accesses per second vs. block size.

Figure 2

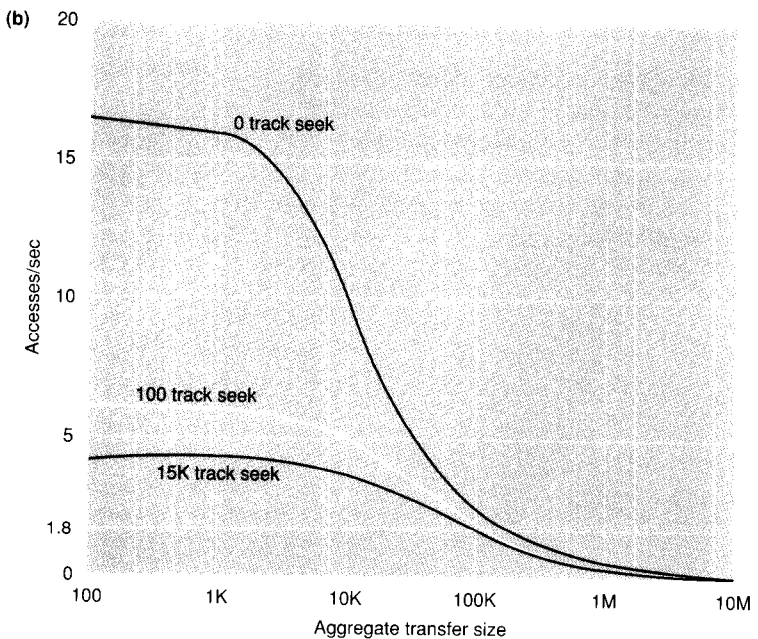
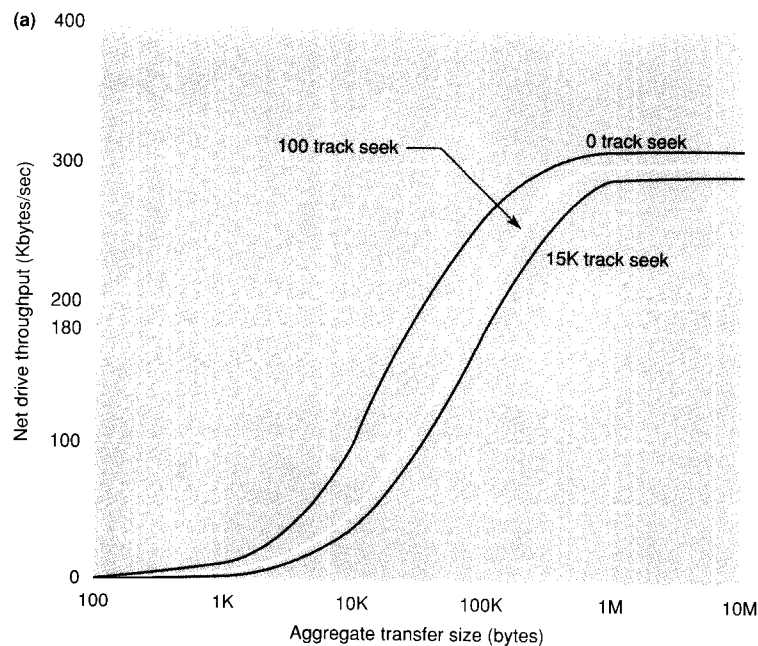


Figure 2. Combined effects of latency, aggregate transfer size, and seek

delay. (a) Net throughput vs. transfer size. (b) Accesses per second vs. transfer size.

Combined Effects of Latency, Transfer Size, and Seek Delays

Figure 2 shows the combined effects of latency, aggregate transfer size, and seek delays on the data throughput and access rate of the drive. Both graphs assume that there is no host overhead to slow down the data transfers, and they assume a constant transfer size of 100 Kbytes.

All the curves assume an average 50 ms rotational latency and 7 ms address verification delay. Note that each curve is labeled with a different seek size. The 15-Kbyte track seek assumes a 190 ms seek delay, the 100 track seek assumes a 100 ms seek delay, and the 0 track seek assumes no seek delay.

The 0 track seek curves in Figure 2 show an absolute upper bound on subsystem performance that can be approached only if the application maintains the physical layout of the data on the media to guarantee locality of reference, for example, an application that does purely sequential writes and reads.

The 15-Kbyte track seek curves are fairly good approximations for subsystem performance when random accesses are being made to a cartridge already loaded in a drive. In Figure 2a, this curve shows that for a transfer of 100 Kbytes, the expected throughput is about 180 Kbytes per second. At that rate, for example, 1.8 bit-mapped images can be transferred in 1 second:

$$\frac{180 \text{ Kbytes/sec}}{100 \text{ Kbytes/image}} = 1.8 \text{ images/sec}$$

This result can also be read directly from the 15-Kbyte track seek curve in Figure 2b.

It should be clear at this point what the drive itself can do. The next consideration is the effects of the jukebox disk changer performance.

Jukebox Performance Issues

While drive performance is determined by familiar parameters like seek time, latency, data rate, and block size, the performance of the jukebox is determined by less familiar parameters—spin-up and spin-down times, eject and insert times, flip times, and elevator motion times. (See the preceding article by Anand Patel for a description of the jukebox and its physical components.)

Performance Profile

Figure 3 shows the average request service time an application would see as a function of the rate at which it sends requests to the jukebox.¹ This curve assumes the following:

- The average platter change time is 16.8 seconds.
- Each piece of data requested is a fixed size (100 Kbytes is a typical number for an image application).
- Requests are randomly distributed across all platters in the box.
- Requests for each platter are queued up as they arrive and can all be serviced when the platter is finally loaded into a drive.

With regard to the last assumption, a few key characteristics should be kept in mind. First, the optical disk process (ODP) sorts all requests for data according to platter side, so each request will be placed in one of 64 logical queues. Platters with many outstanding requests will be guaranteed at least 20 seconds of service time before the platter is removed from the drive. Platters with few requests may be changed in as little as 2 or 3 seconds. The service algorithm guarantees that no request will be "starved" because of high request rates on other platters.

The profile in Figure 3 can be divided into three distinct regions: single request servicing, queue servicing, and saturation.

Single Request Servicing. The jukebox has an average change time of 16.8 seconds, or 3.6 changes per minute. As long as the request rate is less than the maximum change rate and the requests are randomly distributed, each request will usually be serviced by loading a new platter. In this region the average platter change time dominates all other time considerations, so drive performance variables can be ignored.²

Applications with low access rates (less than 3.6 requests per minute per jukebox) that also need to get at random records quickly will operate in this region. An example of an application using single request servicing is described in the inset on page 50.

Figure 3

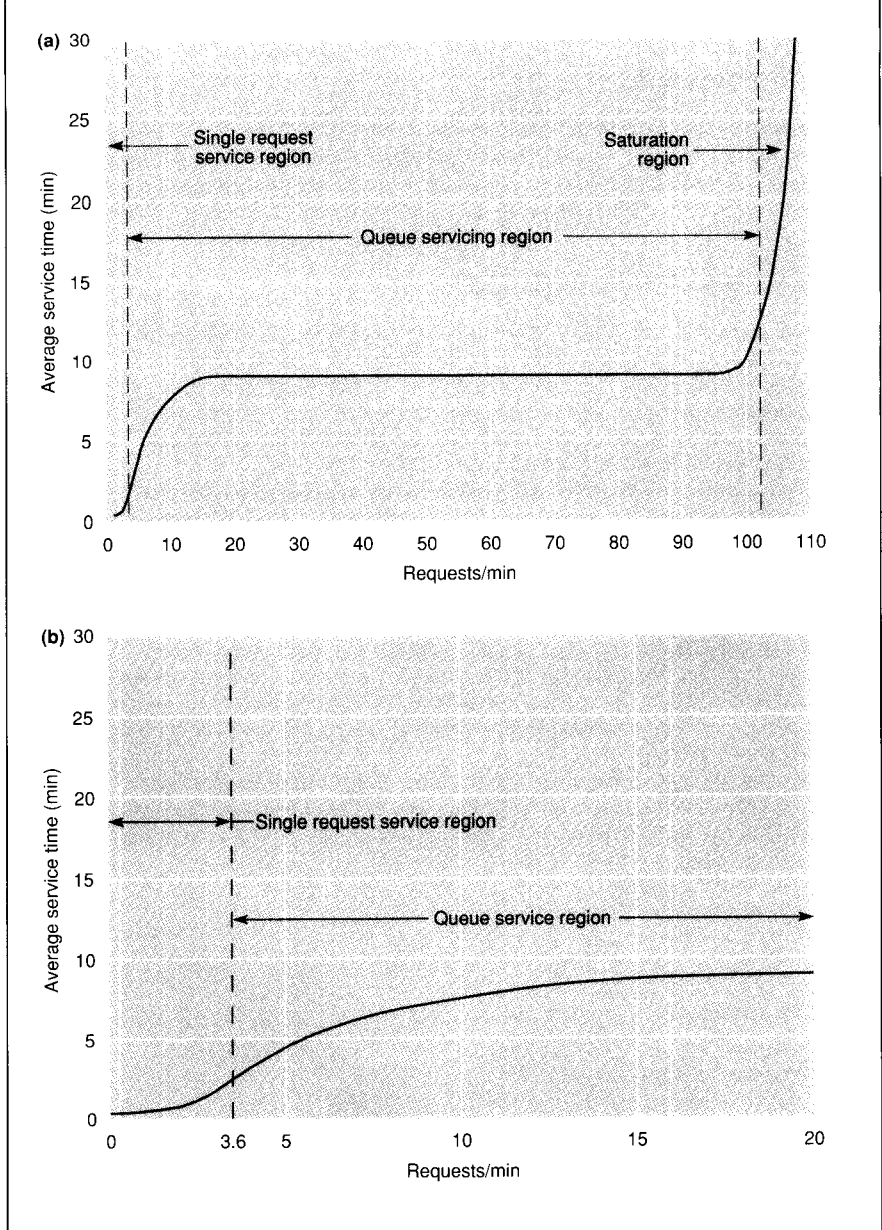


Figure 3. Jukebox performance profile. (a) Average jukebox service time vs. data request rate. (b) Single request service region.

¹In Figure 3, the shape of the curve in the single request service region was derived mathematically. The remainder of the curve was derived heuristically and provides only a qualitative indication of actual service times. The queue servicing region shows the service time rising from about 1 minute to a maximum of 10 minutes. As the system approaches the saturation region, the service-time graph shows exponential growth.

²In the single request servicing region of the graph, most platter changes will result in only one data transfer. In 16.8 seconds the drive can transfer over 5 Mbytes of data. As long as transfers are smaller than that, drive performance has no impact on subsystem performance in this region.

Application Example: Single Request Servicing

Application Assumptions

Single request servicing occurs when the request rate for data transfers is less than the average platter change rate, and when requests are randomly distributed across the platters in the jukebox. It also usually implies that requests are handled on a first-come-first-served basis. One example is an insurance company's customer service center.

Customer Service Center Application

One hundred operators take phone calls requesting information about insurance claims. The claims information is held in a 1000-Gbyte database that could be contained in 12 5200 OSFs. If the average call lasts 3 minutes and the claims information can be retrieved with a single platter mount, the operators require an average platter mount rate of 33 per minute. This is well within the maximum aggregate change rate of the 12 jukeboxes ($12 \times 3.57 = 42.8$).

If claims information for a single customer is spread across platter sides, the "request rate" value would have to be increased to reflect the average number of platter sides that would be used to answer a phone query:

$\text{Change_rate} = 33 \text{ requests/min} \times \text{side_changes/request}$.

There is a certain probability that a request will be made for an already mounted platter that reduces the change rate, but since there are 768 platter sides in this 1000-Gbyte database, the probability is less than 4%. However, if requests are not randomly distributed (i.e., there is some "clumping" of requests for a subset of the platters), there could be an increase in the number of queries that could be serviced by the storage subsystem.

The key contributions of the jukebox in this kind of application are that over 20,000 customer service queries can be handled in a day, each one requiring a single phone call. This can result in improved employee productivity as well as increased customer satisfaction.

Before discussing how the jukebox can easily service request rates much higher than 3.6 requests per minute, it is important to note that even at this comparatively low request rate, the jukebox will operate at its maximum change rate (assuming the requests are randomly distributed). The inset on the following page discusses changer duty cycle and failure rates of the jukebox.

Queue Servicing. Queue servicing is the typical operating mode of many applications. The key characteristic of this region is that the subsystem has a varying resource utilization rate. A cartridge is loaded into a drive, one or more requests are serviced, and then the drive remains idle until the changer can get another cartridge. If request queues have only one request in them, the utilization rate of the drive is about 3%—it spends only about 0.5 second servicing the request and then sits idle until the platter change completes for the second drive. As the queues get longer, the utilization rate increases significantly.

The main advantage of servicing queued requests is that many requests can be serviced for a given platter change, thus the delay in getting to a specific platter can be amortized over many data transfers. This is in contrast to the single-request servicing mode where each data transfer incurs an average 16.8-second platter change delay. For a 100-Kbyte transfer in single service mode, the net data throughput rate for the jukebox is about 6 Kbytes per second. However, if 30 requests were queued up for the loaded platter, all 30 transfers could be made during a platter change time, yielding an average throughput of about 179 Kbytes per second.

The queue servicing mode of operation begins when the request rate becomes greater than the changer rate and requests queue up, waiting for the appropriate platter to be loaded. The average length of the queues depends on the degree to which the request rate exceeds the changer rate. Figure 3 shows the boundaries of the queue servicing region. The left boundary is determined by the platter change rate of the jukebox; the right boundary is determined by performance characteristics of the drive, regardless of the performance of the platter changer.

Randomly distributed 100-Kbyte blocks of data can be accessed at the average rate of 1.8 per second, or 108 per minute. (Refer to Figure 2b.) This average rate of servicing requests depends only on the performance of the drive and can be sustained as long as there are requests in the queue for the loaded platter. Once the queue for the first platter has been drained, the 108-requests-per-minute rate can be maintained by switching to the second drive and servicing queued requests for the platter in that drive. While the second drive is transferring data, the platter in the first drive is changed, so the changer time overlaps with data transfer time. As long as the queued requests for the second platter are not exhausted before the first platter is exchanged, the maximum service rate of the drive can be maintained.

Since arrival patterns of requests are statistical in nature, queues will vary in length. Normally, queues will not be filled enough to use the entire platter change period except in the saturation region.

The following calculations characterize three important operating parameters of the system as it approaches saturation: queue depth, outstanding request count, and request rate. At saturation, queue depth is equal to the number of requests that can be serviced on one platter while the changer is exchanging the other platter. Using randomly distributed 100-Kbyte blocks, with an average access rate of 1.8 requests per second, the depth of the queue required to use the entire platter change time can be computed:

Que_depth

= access rate \times changer time
= 1.8 accesses/sec \times 16.8 sec
= 30 accesses
= 30 requests

Note that access rate and request rate are two different things; access rate is a functional characteristic of the drive and request rate is a characteristic of an application. However, access rate is determined by request rate—indeed it is equal to request rate up to the saturation point. Because of this, average queue depth can be computed as follows:

Que_depth

= request rate \times changer time

Jukebox Duty Cycle and MTBF

The jukebox automatic disk changer was designed to have a 16,000-hour mean time between failure (MTBF), independent of its duty cycle. This means that the jukebox can be used in applications that require frequent changes over extended periods of time. Since the failure rate is totally dependent on power-on hours, the number of changes between failures will vary proportionately with duty cycle.

For example, an application that changes cartridges at the changer's maximum rate, 24 hours per day (100% duty cycle) would average one changer failure in 3,400,000 changes. An application that changes cartridges at half the maximum rate for 12 hours per day (25% duty cycle) would average one failure in 850,000 changes. It is important to note, however, periodic replacement of parts that are subject to wear must be scheduled every 360,000 changes.

The MTBF of the jukebox (including two drives, formatter, and changer) is approximately 5700 hours; the subsystem (including the host controller) has an MTBF of approximately 5000 hours.

Given an average queue depth of 30 requests, the number of outstanding requests would be:

$$\begin{aligned} \text{Req_out} & \\ &= \text{que_depth} \times \text{number of queues} \\ &= 30 \text{ requests} \times 64 \\ &= 1920 \text{ requests} \end{aligned}$$

Saturation. Saturation is reached when the subsystem receives requests at or above the rate it can potentially service them. At that point unbounded queue growth begins. If the system is operated in the saturation region, it is important to understand that while the saturation region yields maximum subsystem throughput, average service times for requests can become very long. (Refer to Figure 3 to see how average service times rise significantly as saturation is reached.) It is also important to configure the system so that it is only in saturation for limited periods of time.

Optimization

There are several strategies for optimizing the use of the jukebox in a variety of settings. In this context optimization can take two forms: maximizing throughput or minimizing access time.

Slot Selection

If requests are not really randomly distributed, frequently used platters should be placed at the bottom of the carousel to minimize platter change time. This reduces the distance the platter has to be moved and can save an average of 1 to 2 seconds per change. Also, if multiple jukeboxes are used, spreading the most frequently used platters evenly among the jukeboxes cuts down on changer contention.

Intelligent Grouping of Data

Perhaps the single greatest variable affecting data access performance is the physical arrangement of the data on the platters. Having frequently referenced data on one piece of media can significantly reduce the number of platter changes necessary and can dramatically reduce average wait times. Putting frequently accessed records together on adjacent tracks can also improve performance.

Fill the Disks

For most applications, having disks completely filled with data reduces the number of disk changes to access the database. However, if the application requires "file growth" or record updating, it may allocate space on the disk to add information later. This initially leads to more platter changes (because more platters are needed to accommodate the database), but it may have significant advantages later on as additional records fill the space and keep related data in close proximity.

Prefetch Data

If there is locality of reference, it is possible to read large blocks of data (up to 5 Mbytes) and cache it on magnetic disk. This is effective since the single service mode allows 16.8 seconds to service a single request. If the system can handle the added data traffic, a 5-Mbyte read can be completed at the same average rate as 10-Kbyte reads.

An example of this approach is a bank loan processing system. When a loan file is being processed, there are several pages of the file that are predictably used together. As soon as the file is opened, a number of pages are read. If the pages are stored as images, it would be possible to retrieve 50 or more pages in a single platter change time.

Note that this suggestion works for the single service mode of operation, but not the queue servicing mode. In the former, a single data block is read and then the drive is idle for most of the changer time. This time could be used to prefetch a large block of data. In the queue servicing mode, multiple blocks are read from the drive which, in itself, may take much of the changer time.

Flip Platters

The queue servicing algorithm will always service requests on the second side of a loaded platter before putting that platter back in the disk slots. This saves an average of 2 to 4 seconds since a flip is faster than any exchange.

Weighted Service Algorithm

Certain applications may have most of their requests going to a small number of platters, with a small number of requests going to all the other platters. This can result in the busy platters not getting enough service time (under the 20-second limit), while on the whole the subsystem is underutilized. The application can circumvent the 20-second rule by implementing its own queuing algorithm and issuing requests to the ODP in a controlled manner.

Handling small block accesses on the jukebox will result in very low average throughput. The overhead of getting to the first byte of a block of information is so significant, it must be amortized over a large number of bytes to yield reasonable throughput.

The performance characteristics of the jukebox are ideally suited for data blocks in the range of 10 Kbytes to 100 Kbytes. When block sizes need to be much larger than 100 Kbytes, the transfer rate of the drive becomes a more significant limiting factor. For block sizes smaller than 10 Kbytes, the transfer rate is not a concern, but the overhead of getting to the data becomes the major performance parameter.

Steve Coleman joined Tandem in 1980 and managed the early phases of development of the 5200 OSF. He is currently doing information and image management investigations.

Conclusions

Although this article is primarily intended to help systems analysts compute sizing and performance parameters, a few general conclusions can be drawn.

Large capacity jukeboxes—60 to 200 platters, with high speed robotics (change times under 10 seconds)—can turn out to be much slower than smaller jukeboxes with longer change times. For applications that operate in the single service mode, absolute changer time is only half the story. What is really important is the product of the number of platter sides in the jukebox and the time it takes to change a platter.

For example, the 5200 OSF has a side-count/change-time product of 1075 (64 sides \times 16.8 seconds), which is the number of seconds it takes to exchange 64 platter sides in the jukebox. A jukebox that changes platters in half the time, but has four times the number of platters, would have a product of 2150 (256 sides \times 8.4 seconds). Thus, if the database required 128 platters, four 5200 OSFs could provide access to all platter sides within 1075 seconds. Only one of the larger jukeboxes need be used, but it would take 2150 seconds to access every platter side. In this case, the 5200 OSF provides *twice* the overall access rate of the larger jukebox.

Technical Paper: The OSI Model: Overview, Status, and Current Issues

The Open Systems Interconnection (OSI) model has been under development by the International Organization for Standardization (ISO) and other standards bodies for nearly 12 years. During this time, the standards for the interworking of equipment from different vendors have turned from vague ideas into reality.

This article gives a brief overview of the OSI reference model and highlights some of the most significant OSI issues. There are a number of factors that are likely to have a major impact on the development of OSI standards and their acceptance in the marketplace over the next ten years. Some of these issues are considered together with the benefits that users can expect to see from the adoption of OSI standards. Finally, this article considers how the international standards can be expected to evolve in the future.

The OSI Model

In 1977 ISO set up a new subcommittee to institute the development of communication standards, which previously had appeared on an ad hoc basis to meet specific needs. The subcommittee's task, titled Open Systems Interconnection, was to develop an architecture of standards that could be used to build distributed information systems. At the same time, the International Telegraph and Telephone Consultative Committee (CCITT) began to work with ISO on the application of standards to telecommunication services.

Ten years' work by the ISO and CCITT resulted in the seven-layer reference model for OSI. In addition, many other standards implementing various parts of the model were produced, and more are currently under development.

Purpose of the Model

The problem that the OSI model was intended to solve arises from the wide range of incompatible computer systems produced by different manufacturers. The difficulties of connecting these "heterogeneous" machines so they can cooperate in their information processing functions are immense. Among the differences are word lengths, representations of data types (such as characters, integers, and real numbers), methods of synchronizing processing functions, and approaches to data communications protocols.

The OSI model was developed to enable these different systems to work together using a common set of protocols to communicate and an abstract model of information processing to signal their application processing semantics to each other.

Benefits to the End User

The objective of the work was to provide major benefits to end users of computer systems. As an organization's data processing services grew from a single computer system to multiple distributed systems, the need to connect those systems to provide shared access to information became increasingly important. Also, as data processing services moved away from batch environments to on-line systems, the requirement to provide real-time access to shared information became critical.

This gave rise to a dilemma: Manufacturers provided the means to interconnect their own proprietary systems using their own set of procedures and protocols, but if a user had equipment from different manufacturers, the result would be islands of isolated information processing. Consequently, users became locked into a single supplier and were unable to base their purchasing policy on normal value-for-money considerations.

The aim of OSI is to provide universal connectivity by defining standard mechanisms for the interconnection of heterogeneous computer systems. Thus, the end user is freed from being limited to a single supplier because of communications requirements.

A second consequence of universal connectivity is that, in addition to interconnecting their corporate systems, users can also connect to other organizations regardless of the type or manufacture of equipment. Business communications can be transmitted between organizations more quickly and efficiently, and major areas of business activity, such as invoicing, bill payment, and order entry, can be fully automated.

In addition to enhanced communications capabilities between enterprises, OSI offers the possibility of new types of services that can be provided over networks and accessed using standard protocols. Some of these services, such as electronic mail, have already begun to appear. The number and scope of such services can be expected to grow as users' ability to access them using standard protocols becomes universal.

The OSI Model in Detail

ISO defined a seven-layer reference model. This was used as a structure within which to develop separate standards to address particular aspects and requirements for interworking between heterogeneous systems. The standards defined in each of these layers provide a set of tools that are used by application processes to communicate and share information.

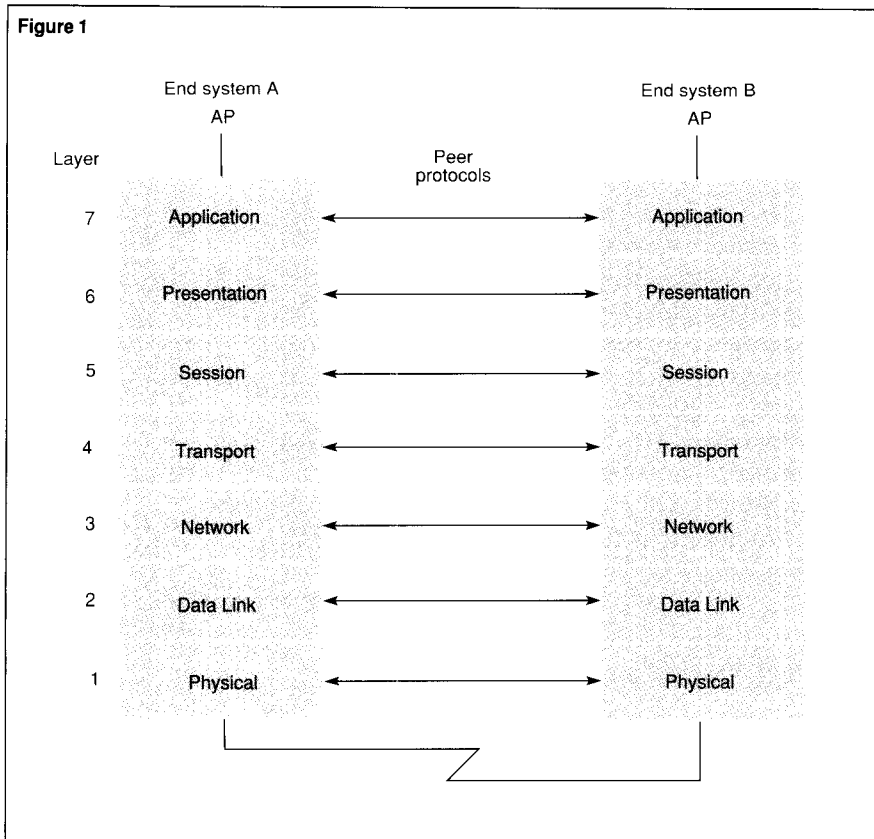


Figure 1.
The seven-layer reference model.

The Seven Layers

The OSI approach to interworking defines an Open Systems Interconnection Environment (OSIE), providing all the functions necessary to allow end systems to communicate. *Application processes* (APs) in each of the end systems communicate through the OSIE and provide end-user business functions. The OSIE includes those networking components that provide the physical connectivity between end systems together with all components within the end systems that provide communication services to the AP.

To provide these services, a wide range of functions needs to be performed, from the physical transmission of the data to the control of dialogs between APs and the physical representation of abstract data types. The OSI seven-layer model separates and groups the functions so they can be better understood and are easier to work with.

As shown in Figure 1, the layers of the model perform the following functions:

- *Application Layer.* Provides the means for the AP to access the OSIE.
- *Presentation Layer.* Provides for the representation of information that applications either communicate or refer to in their communication.
- *Session Layer.* Enables APs to synchronize their dialog and manage their data exchange.
- *Transport Layer.* Provides transparent transfer of data between end systems and a level of reliability independent of the underlying Network Layer.
- *Network Layer.* Provides the routing and switching functions necessary to establish, maintain, and terminate logical connections and transfer data between end systems.
- *Data Link Layer.* Manages the flow of data over physical link(s), including any necessary error recovery and synchronization.
- *Physical Layer.* Provides the mechanical, electrical, functional, and procedural specification for the connections between physical entities.

To provide its service to the layer above, each layer of the model makes use of the service provided by the layer below. Using a peer protocol, each layer also communicates with its peer layer in the cooperating end system. ISO (and CCITT) define two standards for each layer: One standard specifies the service to the layer above, and the other specifies the protocol to communicate with its peer layer in the cooperating end system.

The characteristics of each layer are defined in a modular fashion so that the functions and protocols of each layer can be defined independently. In this way, it is possible to replace the protocol in one layer without requiring changes to any other layer. This leads to great flexibility in the types of networks that can be used and in the range of applications that can be supported. This flexibility provides the capability for OSI to evolve as new technologies and new applications appear.

The seven layers of the model fall into two groups. The lower four layers handle the communication, routing, flow control, and error handling functions necessary to transport data from end to end across the network. The upper three layers handle the coordination of applications across the network and determine the way in which information is presented to them.

At the top of the lower four layers, the Transport Layer is a common link for a range of OSI scenarios. Below the Transport Layer there can be different types of physical networks with differing characteristics. The Transport Protocol brings all of these networks to a single service level and supports them with one Transport Service. Above the Transport Layer there can be different APs providing various types of end-user services.

Table 1 lists some of the more important standards for layers 1 through 6. Many of these standards are also published by CCITT in their X.200 series of recommendations.

Application Layer Standards

The lower four layers of the OSI model define standard mechanisms for the transport of data over a variety of physical communication media. The Session and Presentation Layers define the dialog and syntax conventions that application processes use to communicate in a structured and system-independent fashion. The Application Layer standards are those that actually define the semantics of interactions and enable communication between APs.

The power and flexibility of OSI depend on the range of Application Layer standards, each of which provides the functions required by a particular type of business application. For example, the banking industry needs standards for a range of financial transactions, including electronic funds transfer, ATM, and credit card authorization. In contrast, the airline industry requires standards for making seat reservations, checking baggage, and other transactions associated with running an airline.

In general, each industry will have its own requirements for communication standards between applications that are specific to the industry. There are also needs for non-industry-specific standards such as those for accessing and transferring files of data, submitting jobs, and retrieving output.

Table 1.
Current status of international standards; layers 1 through 6.

Standards organization	Status and standards numbers	Description
General		
ISO	IS 7498	Basic reference model
ISO	IS 7498/AD 1	Connectionless data transmission
ISO	DIS 7498-2	Security architecture
ISO	DIS 7498-3	Naming and addressing
ISO	DIS 7498-4	Management framework
Layer 6		
ISO	IS 8822/8823	Connection-oriented Presentation Service/Protocol
ISO	PDAD 8822-1	Connectionless Presentation Service
ISO	DP 9576	Connectionless Presentation Protocol
ISO	IS 8824	Abstract syntax notation one (ASN.1)
ISO	IS 8825	Basic encoding rules for ASN.1
Layer 5		
ISO	IS 8326/8327	Connection-oriented Session Service/Protocol
ISO	DAD 8326-3	Connectionless Session Service
ISO	DIS 9548	Connectionless Session Protocol
Layer 4		
ISO	IS 8072/8073	Connection-oriented Transport Service/Protocol
ISO	IS 8072/AD 1	Connectionless Transport Service
ISO	IS 8602	Connectionless Transport Protocol
Layer 3		
ISO	IS 8348	Connection-oriented Network Service
ISO	IS 8348/AD 1	Connectionless Network Service
ISO	IS 8208	X.25 packet layer protocol
ISO	DIS 8881	Use of X.25 in ISO 8802 LANs
ISO	DIS 8473	Connectionless Network Protocol
ISO	DP 9068	Connectionless Network Service using X.25
Layer 2		
CCITT	X.25/LAPB	High level data link control (HDLC)
Layer 1/2		
ISO	DIS 8802	LLC and media access control for LANs (see Table 3)
Layer 1		
CCITT	X.21	Interface between DTE and DCE for synchronous data networks
CCITT	X.21bis	Interface between DTE and DCE for equipment using synchronous V-series modems

Refer to the OSI acronym list for full names of acronyms used in this table.

Table 2.
Application Layer standards.

Standards organization	Status and standards numbers	Description
Common services		
ISO	DIS 8649/8650	Association control service/protocol
ISO	DIS 9804/9805	Commitment, Concurrency, and Recovery service/protocol
ISO	PDAD 8649-2	Application-unit-data service (connectionless)
ISO	DP 10035	Application-unit-data protocol (connectionless)
ISO	DIS 9072	Remote operations service
General		
ISO	IS 8571	File transfer, access, and management
ISO	DIS 9040/9041	Virtual Terminal service/protocol
ISO	DP 8831/8832	Job transfer and manipulation service/protocol
ISO	DIS 8613	Office Document Architecture
ISO	DIS 8505/8883/9065/9066/9072	Message-oriented text interchange systems (equivalent to CCITT X.400 series)
ISO	DIS 9594	OSI directory services (equivalent to CCITT X.500 series)
ISO	DIS 9506	Manufacturing message specification
ISO	DP 9579	Remote database access
ISO	DP 10026	Distributed transaction processing
ISO	DP 10031	Distributed office applications
ISO	DP 9595/9596	Management information service/protocol
Banking		
ISO	DP 9955	Methodology for banking standard development
ISO	DIS 8583	Message content for card-based messages
ISO	DP 8732	Key management for message authentication
Business EDI standards		
ANSI	X12.1	Purchase order transaction set
ANSI	X12.2	Invoice transaction set
ANSI	X12.3	Data element dictionary
ANSI	X12.4	Remittance/payment advice transaction set
ANSI	X12.5	Interchange control structure
ANSI	X12.6	Application control structure
ISO	IS 9735	Electronic data interchange for administration, commerce, and transport (EDIFACT)

OSI must provide a variety of Application Layer standards to have wide acceptance for intersystem communications. Table 2 lists some of the important standards that have been completed or are under development. This list continues to grow as more standards committees begin to incorporate their own application areas into the OSI environment. Much of the work is still in its early stages, but a core of standards now exists and products are beginning to appear that implement all seven layers of the OSI model. As new standards reach maturity, the number of products will increase dramatically.

Message Handling Systems

Messaging systems form an important group of Application Layer standards. The Message Handling Service (MHS) enables users—either human or computerized application processes—to exchange messages that include text, graphics, telex, facsimile, voice, or other data.

MHS was originally specified by CCITT in their X.400 series and is now taken up by ISO in their Message-Oriented Text Interchange System (MOTIS) standards. Products for these services are available on a range of hardware, and there have been numerous demonstrations organized at international exhibitions to show interworking between them.

Message transfer is realized by a “store and forward” mechanism that relays information transparently regardless of its data type. CCITT defines protocols for forwarding messages between public messaging services and between public and private messaging services. The ISO MOTIS standards extend this to communication directly between private messaging services.

One of the MHS services defined in X.400 is the Inter-Personal Messaging Service (IPMS). IPMS defines an electronic mail system between individuals registered with an MHS service. Users can compose and send messages to each other and have them delivered to one or many recipients. Messages can include multiple components of different types, and the service can perform automatic conversions between different message types.

MHS standards have acted as a catalyst for other activities because they provide a generic message transfer service not only for electronic mail but for transport of all kinds of data. Therefore, interest has increased in other business-related standards such as Office Document Architecture (ODA) for document transfer and Electronic Data Interchange (EDI) for invoicing, order entry, and other business functions.

Products that implement the MHS standards give a new dimension to communications within an organization and between separate enterprises. Many organizations have had electronic mail systems for some time, but these have been limited in most cases by the incompatibilities between systems and the absence of standards. This has often led to organizations running multiple incompatible mail systems with no possibility of interconnection.

Now organizations can link their messaging systems using MHS standards, and they can extend their electronic communications to other independent enterprises. There are two ways to link messaging systems. The connection can be made either through a direct link or indirectly through a public messaging service. The former might be used where there is significant traffic between enterprises on a regular basis and the latter for a one-time communication.

MHS standards will have a major impact on the way enterprises manage their communications. The paper of today will be replaced by the electronic messages of tomorrow.

File Services

File transfer, access, and management (FTAM) is another important area of standardization and implementation. Attention was initially focused on this area by the General Motors MAP (Manufacturing Automation Protocol) initiative and by Boeing's TOP (Technical and Office Protocols).

FTAM provides three basic capabilities:

- Transfer of complete files of data from one end system to another across an OSI network.
- Access to records within a data file for read, write, or update across an OSI network.
- Interrogation and/or change various file attributes across the network (file management).

To provide these functions, FTAM defines a *virtual file store*, which includes a superset of most of the properties found in typical real file systems today. To make their file service available to the OSI environment, implementations of FTAM must map this virtual file store to their local real file system.

MAP/TOP-compliant FTAM products are now available on a variety of systems, and the use of FTAM for vendor-independent transfer of files within heterogeneous networks is increasing.

FTAM provides the solution to a persistent problem for both users and suppliers of systems over many years. Wherever information needs to be shared or moved between systems, FTAM provides the standard mechanism for doing this regardless of the end systems concerned.

Terminal Services

The Virtual Terminal (VT) standards have been the subject of considerable attention over many years and are now largely complete. These standards provide a mechanism for an application program to drive a terminal over an OSI network. VT defines a generic terminal having properties that can be mapped to a variety of real terminals. Because of the variety of terminal types and capabilities, VT defines a number of terminal classes. These include *basic class*, a simple text-oriented mode of operation, and *graphics class*, a generalized support for graphical display devices.

Although the VT standards are now in place, the interest and adoption of these standards has not been high largely because of the growing use of intelligent workstations in place of dumb terminals. When a terminal is replaced by an intelligent workstation, the screen presentation can be handled locally on the workstation. Communication with remote applications can be achieved with a variety of other OSI application protocols, such as FTAM, or application-to-application communication standards, such as OSI-TP (transaction processing).

Transaction Processing

Transaction processing standards are making rapid progress through the ISO committees. Transactions are characterized by a set of operations (typically on a database) that must be either completed successfully or not at all and that are also independent of all other sets of operations.

Within OSI, the concern is to develop a standard for distributed transaction processing where individual operations span two or more OSI end systems. The standard, called OSI-TP, has as one of its key requirements the management of failures (in the communication or in the end systems) in such a way that the transaction is either fully committed or entirely revoked. To achieve this, OSI-TP uses a two-phase commitment procedure based on the common application service for Commitment, Concurrency and Recovery (CCR), one of a standard set of services available to Application Layer protocols.

The OSI-TP work is progressing quickly, and a full international standard is expected by mid-1990.

Security

As OSI moves from the realm of theory to the real world of business, security has become a major concern for all Application Layer services. This is reflected in the number of committees now addressing security issues. Originally, the reference model did not include any specific provisions for security requirements, and most OSI standards provide very little scope for implementors to include security mechanisms.

One of the first objectives was to develop a security addendum to the basic OSI reference model. This work is now well advanced, and draft documents are available for authentication framework, access control framework, and non-repudiation framework. Existing upper-layer standards will be extended to incorporate this new work.

Directory Services

The work on standards for Directory Services arose largely out of MHS standards. The directory standards have been developed by CCITT in association with ISO and were published in 1988 as the X.500 series of recommendations.

These standards define a structure for a distributed "directory information base" and the means by which this information can be accessed via OSI services and protocols. Using a limited set of known attributes, directory services enable users to find the OSI address of an object or person. With the increasing penetration of MHS services, there will be large numbers of products that implement directory services of this type.

One of the significant motivating factors behind this work was the PTT's (postal, telephone, and telegraph public services) requirement to offer new chargeable services to network users. This resulted in the rapid progress of the work to full standardization. Significant product growth will occur in this area over the next few years.

The Physical Communications Medium

At the other end of the scale from the Application Layer standards are the range of physical media that can be used to build an OSI network. Originally the Network, Data Link, and Physical Layers were specified for wide area networks (WANs) based on the CCITT packet switching standard X.25 with synchronous communication links. Later, work was begun by the Project 802 committee of IEEE (Institute of Electrical and Electronic Engineers) to develop standards for local area networks (LANs) based on a variety of ring and bus topologies. The developing standards were compatible with the OSI reference model and were later adopted by ISO as alternatives for the lower layers of OSI. Table 3 lists the current standards in this area.

Table 3.
LAN standards.

IEEE standard	Status	ISO standard	Description
802.1			Systems management
	DIS	8802/1	General introduction
802.2	DIS	8802/2	Logical Link Control
802.3	DIS	8802/3	Carrier sense multiple access with collision detection
802.4	DIS	8802/4	Token-passing bus access method
802.5	DIS	8802/5	Token-Ring access method
802.6			Metropolitan area networks
	DIS	8802/7	Slotted-ring access method
	DIS	9314	Fiber-distributed data interface (equivalent to ANSI X3.139-1987)

LANs vs. WANs

The important differences between WANs and LANs are that WANs can be distributed over a much larger area, but LANs typically provide much higher data throughput rates. Typical WAN network connections operate at speeds up to 64 Kbits per second, where LANs operate in the 1-to-10-Mbit-per-second range.

This brings possibilities for applications on a LAN that could not be considered on a WAN. To date, LAN services include print servers, file servers, and document servers, as well as remote procedure calls, providing distributed processing functions.

LAN Protocols

To take advantage of the particular characteristics of LANs, such as high bandwidth, high reliability, and short transit delays, a set of protocols have been defined for the lower three layers of the model that differ from the X.25 protocols used in the WAN environment. The IS 8802 series of standards defines the two lower layers of the OSI model for a variety of LAN types listed in Table 3. ISO has produced a new standard for the Network Layer, which is specifically designed for LANs.

Physical Layer: There are a number of possibilities for the physical medium of the LAN. IS 8802/3 describes a baseband bus configuration on which nodes contend for access to the bus to transmit data. This mechanism, known as CSMA/CD (carrier sense multiple access with collision detect), is commonly used with a passive coaxial cable as the transmission medium. Versions are also available using twisted pairs or optical fiber.

In IS 8802/4, access to the medium is controlled by passing a token between the nodes of the LAN. The token confers the right to transmit data. One of the features of this method is the ability to assign higher transmission priorities to some nodes by ensuring they receive the token more frequently. The data is normally modulated onto an RF (radio frequency) channel on a broadband coaxial bus. IS 8802/5 also uses a token passing technique, but in this case, a ring topology is used and active repeaters relay the data from one node to the next around the ring.

Of the other access methods listed in Table 3, the metropolitan area networks (MANs) provide high bandwidth communications over an area covering tens of kilometers and are still under-going definition. The slotted ring method is well defined but has proved less popular than other methods.

Data Link Control. At the Data Link Layer of the LAN, the same procedures are used regardless of the physical access method selected. These procedures, defined by IS 8802/2, provide communication between link stations on the LAN.

Logical Link Control (LLC) Type 1 service is used to provide connectionless data transfer, while LLC Type 2 service is used for both connectionless and connection-oriented data transfer. The connection-oriented service implies that a logical link is made between two cooperating link stations and functions such as error recovery and flow control are performed. With the connectionless service, no logical link is formed, so there is no error recovery or flow control.

Network Layer. The Network Service, which is defined by IS 8348, provides a choice between a connection-oriented protocol (IS 8208) or a connectionless protocol (IS 8473). IS 8208 is similar to the X.25 Packet Level and requires that the underlying Data Link Layer provide a sequenced, error-free transfer of data. This could be used with 8802/2 LLC Type 2 but is not suitable for use with LLC Type 1. The connectionless protocol IS 8473 must be used with LLC Type 1.

Transport Protocol Class. To provide the necessary quality of service required by the higher layers, the Transport Service must support those functions that are absent from the lower layers. In connectionless lower layers, the Transport Layer must provide functions such as sequencing, flow control, error detection, and recovery. This is achieved using a Transport Protocol known as Class 4.

There are four other classes of Transport Protocol that can be used to provide the Transport Service to the higher layers. The choice of which class to use depends on the type of underlying network and functionality required. Where connection-oriented lower layers are used on the LAN, Transport Protocol Class 0 can be used. This assumes that the error recovery and flow control are provided by the underlying layers. To date, most implementations of OSI on LANs have used connectionless Data Link and Network Layers in conjunction with Transport Protocol Class 4.

New LAN Technology

The next generation of LANs is now beginning to appear. ANSI (American National Standards Institute) is working on a fiber-distributed data interface (FDDI) network that operates at 100 Mbits per second. The 802.6 committee of IEEE is beginning to make progress with its MAN architecture, which will extend the reach of high-speed networks to an area the size of a city.

High bandwidth opens up the possibility for true distributed processing with individual operating system functions allocated to dedicated servers on a LAN. Many new applications, such as interactive voice and video, are also possible.

With the advent of these new services, it will be necessary to extend the OSI standards throughout all the layers of the reference model. The reference model itself will need to evolve in order to incorporate new technologies and services, and some consideration is given to the possible direction of these changes later in this article.

Internetworking

The two types of physical networks that OSI standards have defined so far specify different sets of protocols in the lower three layers of the reference model. IS 8473, sometimes referred to as the "Internet Protocol," defines a connectionless Network Layer for the LAN environment. In a connectionless service, each data unit is sent independently without prior connection establishment and without error detection or recovery. This form of data unit is sometimes referred to as a *datagram*.

Figure 2

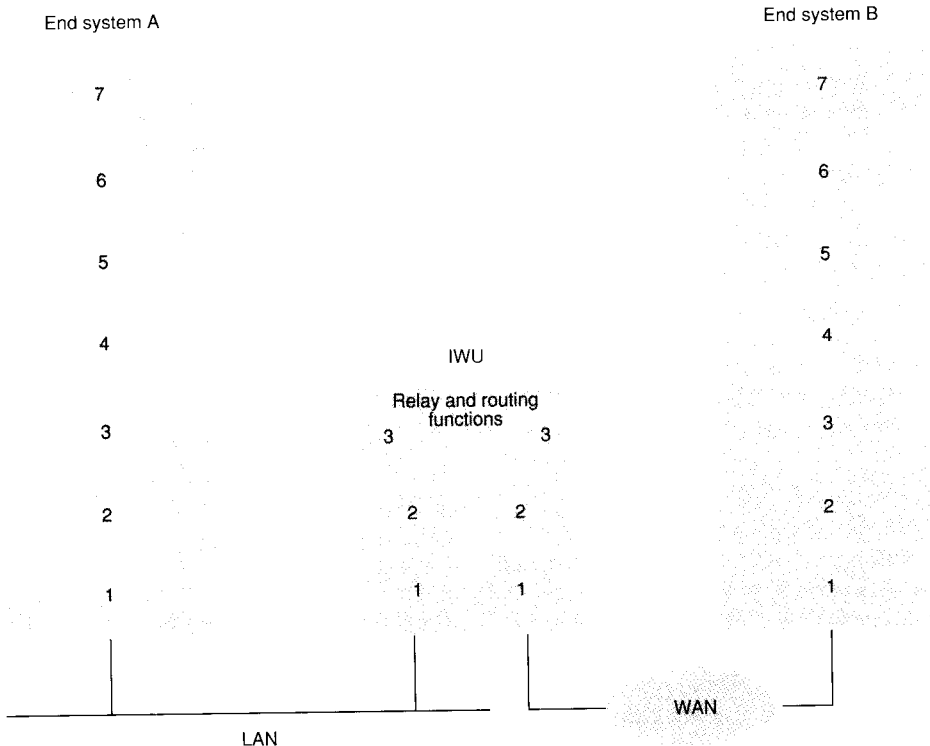


Figure 2.
Internetworking between
LAN and WAN.

In the WAN environment, the Network Layer is defined by X.25. This is a connection-oriented protocol in which data is packetized for transmission on virtual circuits. There are three phases of communication: connection establishment, data transfer, and connection release. Data packets can only be transmitted while the connection is in the data transfer phase.

When spanning more than one physical network or subnetwork, the OSI communication requires an *interworking unit* (IWU) to provide the relay and routing functions between adjacent networks. In the OSI model, communication between two end systems requires an IWU operating at the Network Layer. (See Figure 2.)

This model works well with similar networks, such as a public and private X.25 network, but there is a problem when the networks use different Network Layer Services, as is the case when one is connectionless and the other connection-oriented. According to the reference model, relaying cannot be performed above the Network Layer, so the ISO committees are considering several possibilities.

One proposal is to use the X.25 Network Protocol over the LAN, providing connection-oriented services in the LAN environment. Another extends the connectionless Internet Protocol to the WAN. Both of these approaches are the subject of approved ISO standards, but LAN suppliers do not view the former favorably as they prefer to use connectionless services. The latter proposal has a similar problem with the WAN suppliers.

ISO and other standards bodies are considering a more radical solution that would extend the OSI reference model to allow relaying at the Transport Layer. This would permit the use of dissimilar Network Layer Services on the adjacent networks so that a connectionless Network Service on the LAN could be relayed to a connection-oriented service on the WAN.

This problem has not become a serious issue so far, but as use of OSI across multiple networks grows, a rapid resolution will emerge. The problem will be resolved either by an adjustment to the model or from a de facto adoption of one of the two alternative approaches.

Connectionless Services

The reference model, as originally defined, specifies connection-oriented services for all layers of the model. With the appearance of LANs, which typically use connectionless services at the Network Layer, there was a need to extend the reference model to allow new types of services and protocols. ISO has therefore extended the reference model (IS 7498) with an addendum defining connectionless modes of operation. ISO also defined connectionless protocol and service standards for the Network Layer in line with the reference model extensions.

Although connectionless services are used at the Network Layer, most of the higher layer protocols and services use connection-oriented modes of communication. This is possible because Transport Protocol Class 4 has the ability to provide a connection-oriented Transport Service over a connectionless Network Service.

However, ISO has recognized the need for applications that use connectionless modes of communication and is now extending Transport, Session, and Presentation Layer standards with connectionless modes of operation. Applications that use this type of service might be, for example, broadcast services or applications requiring short duration transmissions on an infrequent basis.

Although connectionless addenda to the upper-layer standards are largely in place, specific Application Layer Services have yet to be defined.

OSI Management

ISO is also extending the reference model to include network management. This area is becoming a major concern to network suppliers and operators as increasing complexity demands powerful tools to observe and control network components and functions.

Management Framework

Currently, ISO is defining a set of OSI Management standards to form the basis of tools for these functions within an OSI network. The OSI Management framework standard has four components:

- Common Management Information Services (CMIS) and Common Management Information Protocols (CMIP).
- Specific Management Information Services (SMIS).
- Structure of Management Information (SMI).
- Directory Services.

CMIS/CMIP. These define the fundamental services provided by the management model and the protocols used to transfer management information between end systems. They provide for event reporting, information transfer, and control functions relating to the objects stored in a *management information base* (MIB). The MIB is used to store details of all objects within an OSI management domain.

SMIS. These provide services for specific management functional areas. Those currently defined are:

- Fault management.
- Security management.
- Configuration and name management.
- Performance management.
- Accounting management.

Work is proceeding on these specific services, and there are draft standards for each of them.

SMI. The SMI defines the syntax and semantics for the information in the MIB. The MIB is a distributed database containing details of all objects of concern to the OSI management environment.

Directory Services. The Directory Services standard defines services to perform mapping between names of entities as known by users and addresses of those entities within the OSI environment. This work is based on extensive work carried out by CCITT and published as their X.500 series of recommendations in 1988. The ISO standards are at an advanced stage and will be fully compatible with CCITT.

Future of OSI Management Standards

OSI Management standards are at a relatively immature stage, and despite strong interest from network users and suppliers, the process of agreeing on a common approach requires considerable time. It may well be several years before a full set of standards is in place.

In an attempt to speed up this process and promote the use of OSI Management standards, a group called the OSI Network Management Forum was set up in 1988. Made up of network and computer suppliers, the purpose of the Forum is to encourage industry to "conform to standards in a uniform way."

The Model in Practice

Having looked at the basic building blocks of OSI and how some of them are developing, it is useful to consider an example of how the standards are used to solve a specific set of problems. General Motors made one of the first initiatives in the application of OSI standards when the company set up a group of users to specify protocols for factory floor automation in automobile manufacture.

These Manufacturing Automation Protocols (MAP) define a selection of OSI standards for this environment. Since the complete set of OSI standards was not in place when the first version of MAP was defined, the current draft standards were used where available, and application-specific protocols were defined where no draft was available. Later versions of MAP incorporated the full OSI standards as they became available.

MAP was one of the first user-driven initiatives aimed at establishing a complete seven-layer set of standards for a particular application area. As large numbers of users added their support and requested product compliance, MAP rapidly became one of the major forces driving the implementation of OSI standards.

Elements of MAP

GM selected OSI because they needed to be able to procure equipment from different suppliers and integrate it on the factory floor for their manufacturing operation. The different components of MAP were each chosen to fulfill particular requirements of this operation.

Application Layers. One of the requirements is to manage the operation of robots on the production line. Each robot needs to have the correct program loaded for the current production line job, and any changes to its work schedule require a new program to be loaded. The file transfer capability of FTAM enables programs to be transferred between the control systems and the production line.

Since FTAM provides greater capability than simple file transfer, the early versions of MAP (2.0 and 2.1) specified only a subset of FTAM required for file transfer. The latest version (3.0) specifies the complete standard.

Another requirement is the management by control systems of shop floor devices in real time, including monitoring and controlling a production line and its supply of parts. OSI did not originally specify a standard for this task, so GM defined a manufacturing message format standard for this purpose. This standard has now been adopted by ISO as the Manufacturing Message Specification (MMS) and is in the process of being defined as an OSI standard.

In the absence of OSI standards, MAP originally defined proprietary network management and directory services. However, as the ISO standards become available, they are being incorporated into the MAP specifications.

Lower Layers. There are a number of factors influencing the choice of physical connection requirements for MAP. Since all communications take place within a small geographical area, a LAN is the obvious choice. The choice of LAN type is influenced by several factors: the type of cable already installed, the accommodation of different communication channels on the same wire, flexible topology for the configuration of the network, and resistance to electrical interference. These factors resulted in the choice of a broadband coaxial LAN for the physical medium in MAP.

The access mechanism selected for the broadband LAN was a token-passing bus. In this method, the LAN stations pass around a token that carries the permission to transmit data. This method provides guaranteed maximum delivery times on the LAN, as opposed to the CSMA/CD method, which uses contention and provides nondeterministic delivery times. The provision of guaranteed delivery times is important for real-time operations in the factory.

The broadband token-passing bus was originally defined by IEEE and has now been adopted by ISO as IS 8802/4.

Layers 2 through 6. MAP specifies the connectionless Data Link Layer, LLC Type 1, together with the connectionless Internet Protocol at the Network Layer. Above the Network Layer, MAP specifies Transport Protocol Class 4 and the OSI Session Layer. Early versions of MAP (2.0 and 2.1) did not use a Presentation Layer, since it was not fully defined at the time. The current version (3.0) now includes the OSI Presentation Layer.

The MAP initiative was one of the major factors in promoting the use of OSI for real applications and was the first of a number of initiatives extending the usability and acceptance of OSI standards.

Selection and Testing of Standards

The example described above shows how standards are used to perform application functions over a broadband token-passing bus. FTAM is one of the Application Layer standards specified by MAP, and this makes certain demands on the underlying layers. For example, the Session Layer uses a group of functional units associated with the synchronization service. A different Application Layer (e.g., MHS) would use a different group of functional units. (MHS uses the activity management functional units.)

Similarly, if an X.25 packet switching network were used instead of a broadband token-passing bus, connection-oriented protocols would be used in the lower three layers, and Transport Protocol Class 0 would be used instead of Transport Protocol Class 4.

Profiles and Functional Standards

There are now a considerable number of OSI standards addressing all seven layers of the model. Many of these have subsets and options within them. User and implementor groups realized that if implementors selected different sets of standards or if they used different options within a standard, their products would not communicate.

A number of user and implementor groups, as well as government bodies, became active in specifying sets of standards, or "profiles," and the options to be used within them for particular application areas. Implementors also recognized that some standards contained errors, were ambiguous, or left certain details unspecified. To resolve these issues, they defined "functional standards" for use in various application contexts.

Figure 3

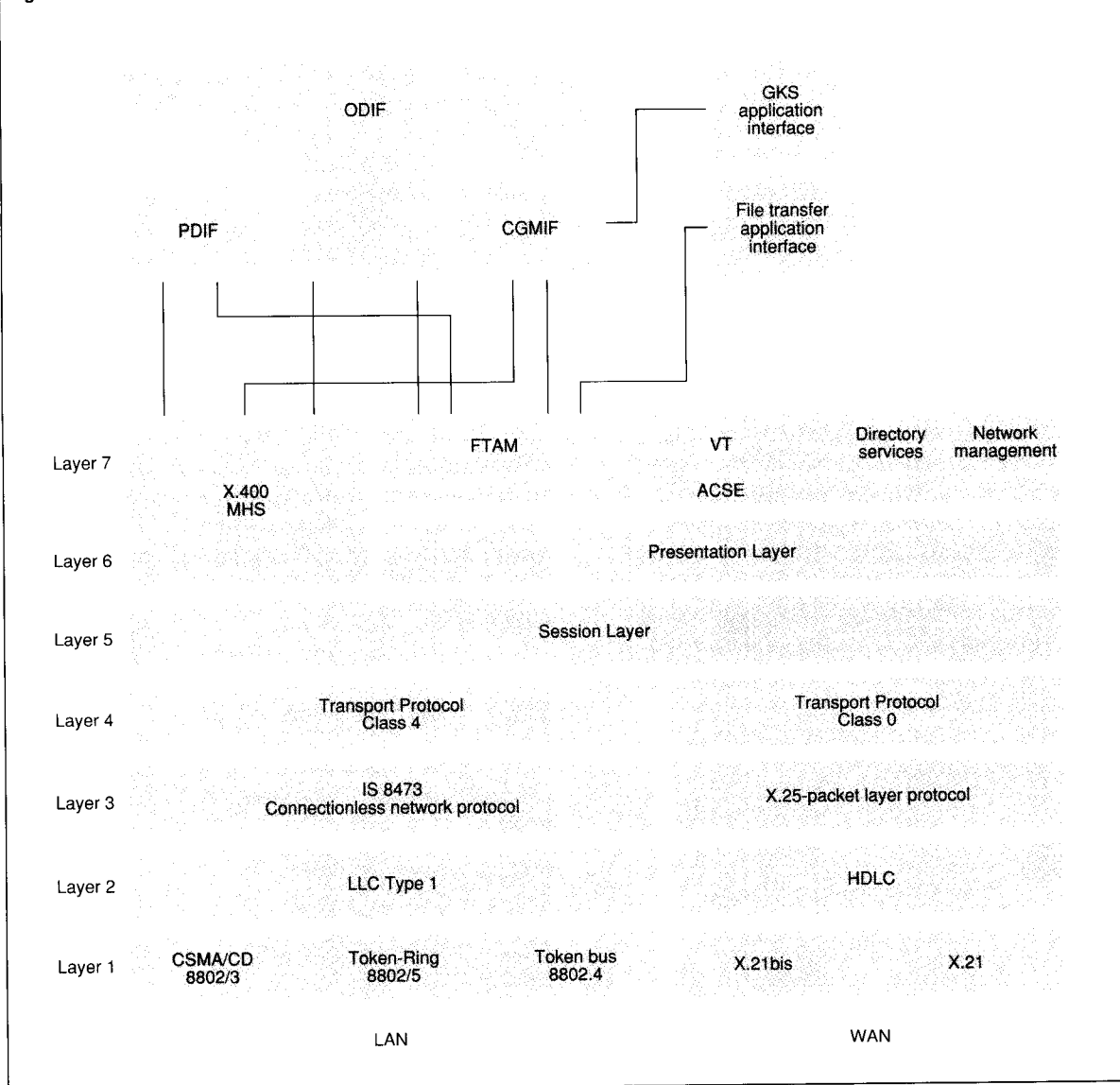


Figure 3.
The TOP standards. Refer to the OSI Acronym list for full names of the acronyms used in this figure.

MAP One of the first profiles defined was the MAP specification described above. MAP specifies protocols for use in the manufacturing environment.

TOP A similar group sponsored by Boeing defined TOP for office systems. Using a LAN environment, TOP specifies a baseband coaxial bus as the medium and CSMA/CD for the access method. TOP also provides for use of an X.25 Network Layer as an alternative to a LAN. Figure 3 shows the protocols and standards specified in TOP, including the two alternative networks and the higher layer standards.

Application Layer services include FTAM, MHS, and VT, as well as directory services and network management. Figure 3 also shows a number of applications that use FTAM and MHS services. These provide document, graphics, and file transfer services for the office environment.

Government Standards Bodies. The National Institute for Standards and Technology (NIST, formerly NBS, or National Bureau of Standards) in the U.S. and the European governmental standards body CEN/CENELEC¹ have been active in specifying functional standards to define exactly how suppliers should implement their OSI-based products. Other important specifications are the government OSI profiles (GOSIP) in the U.S. and U.K. The latter, GOSIP in the U.K., specifies mandatory standards for use in all government-procured contracts.

Conformance

Defining a set of compatible standards is not sufficient requirement to ensure successful interworking between systems. If end systems are to be able to communicate without prior agreement or arrangement, there are many issues relating to the details of the communication that must be first resolved. The objective for open systems interworking is that only address information and security authorization need be exchanged before communication can take place.

¹European Committee for Standardization/European Committee for Electrotechnical Standardization.

One way to improve the probability of interworking successfully is to test intended implementations against a conformance testing center to ensure compliance with the standards. Several groups have been active in setting up testing centers. These include the Corporation for Open Systems (COS) in the U.S. and the Standards Promotion Application Group (SPAG) in Europe. In addition, an initiative called Conformance Testing Services (CTS) in Europe helps to ensure a common testing environment. The aim is that all testing services in Europe will produce the same results on the same set of software.

The main areas covered by the current generation of testing services are FTAM, MHS, and the lower layers of the OSI model. In the future, users will expect products to have a certificate of conformance from one of the official testing centers. These products will have a much higher probability of interworking with other products and, therefore, will be more attractive to prospective customers.

User Benefits from OSI

Users can expect many advantages when they use OSI products. These benefits fall into three areas:

- Multivendor networks.
- Use of public (or semi-public) services.
- Communication between independent enterprises.

Multivendor Networks

Increasingly, businesses are placing more of their critical business functions on information processing systems. It is now commonplace for information vital to the operation of a business to be stored and processed on computers. Communication services provide access to the information for individuals who require it.

To provide the best possible systems on which to store, process, distribute, and present the information, it is necessary to have flexibility in the choice of equipment. Only in this way can a business remain competitive in the way it manages its information.

For maximum flexibility, users need to select equipment that offers the best price/performance and easily integrates into the total information processing environment. Ideally, equipment from different manufacturers would be as easy to integrate as that from the same manufacturer. In practice this is difficult to achieve, but only by striving for this objective can users hope to choose the best equipment for the job.

OSI standards attempt to solve this problem. By ensuring that all equipment conforms to the same standards, it is possible to build a network with different types of systems working together in a single distributed processing environment.

The benefits for the user are, therefore, the ability to interconnect systems from different vendors, the ability to choose the best solution for an application (without having to be concerned about interworking issues), and the ability to replace systems when more attractive alternatives become available.

Public Services

With the advent of public data networks in the 1970s and, later, privately managed networks offering access to value-added network services, the possibility arose to make services available to a range of users. These services include electronic mail, database access, and a variety of information services.

Initially the services were accessed using simple terminal protocols, such as the CCITT standard X.29 over an X.25 carrier network. However, as services became more sophisticated, new protocols such as X.400, Teletex, and EDI were used to access them. Clearly, if service providers were to specify their own access protocols, it would be impossible for a user to access more than a handful of services, and these at considerable cost and effort.

Standards, and OSI standards in particular, make it possible to access a range of services by installing a small number of standard protocols on the access system. Furthermore, MHS standards allow private messaging networks to be linked to public messaging networks and, therefore, to global networks providing access to businesses on a worldwide scale.

Only by using standards can a user have access to the growing number of data services now being offered. In many countries, government regulations specify that services must be available via OSI standards, so OSI will become mandatory for access to these services.

Communication between Independent Enterprises

One of the objectives of OSI is the ability to achieve communication between systems without prior arrangement or negotiation. OSI systems should be able to communicate in the same way that a telephone is used to dial any other telephone in the world without prior agreement (other than knowledge of the telephone number).

Directory services and security features are two important aspects of OSI work that will facilitate communication without prior agreement. Security features are being added to control access and prevent unauthorized use of information and resources. Directory services allow users to find the address of the service they require.

An open network can be used for a variety of purposes. For example, EDI standards can be used to send orders, invoices, or payments directly between suppliers and customers. Electronic business communications can include documents, voice, and video. Businesses can offer chargeable information or data processing services, and users can access them with a minimum of formality. Such communication without prearrangement is possible only if equipment throughout the world conforms to the same standards.

OSI standards will create a new industry centered around providing open communications between enterprises. If they are to take advantage of the new opportunities, businesses will have to comply with these new global standards.

Users' Product Requirements

If users' demands for OSI arise from the considerations described above, there are considerable implications for the type of products they will need from manufacturers and suppliers.

To build multivendor networks, it is not sufficient for equipment to provide a minimum level of OSI support. OSI products must be fully integrated into the system. This means that systems must make available their full range of facilities and services to the OSI network. For example, the FTAM service interface can only be used to maximum advantage if there is software to map the FTAM virtual file-store to the real file-store of the system.

Today there are examples where the lower levels of OSI are used to transport data between end systems, but proprietary protocols are used for the higher layers to implement a proprietary network architecture. This approach places constraints on the openness of a network and limits the flexibility when mixing different types of equipment. This is sometimes unavoidable when OSI upper layer standards are not available to provide the required functionality. However, as the scope of the standards widens, the need for such proprietary protocols should diminish. It will be possible for manufacturers to move their own network architectures toward exclusive use of OSI standards.

From a user's point of view, such changes are an advantage because they increase the flexibility to choose the best hardware and software solution for the application. Since vendors' products have different performance and functional characteristics, it is a considerable advantage if a user can select products that best meet the requirements. This is not possible when issues of integration with existing systems limit the choice. Users need to have products that offer their full range of benefits through the OSI interface.

The Future of OSI

With the approach of the 1990s, the growth of the OSI-based products and services will begin to accelerate. The years of investment by manufacturers, users, and public bodies will begin to bring returns. At the same time, the development and evolution of standards will continue at an accelerating pace.

OSI Networks and ISDN

Further developments in technology will have a profound influence on standards. The Integrated Services Digital Networks (ISDN), which have been under definition by CCITT for most of the 1980s, will come into existence. These networks integrate all forms of electronic communication using digital switching techniques, and as a result, PTTs hope to replace existing voice and switched data networks by these services over the next ten years.

New mechanisms will be needed to migrate distributed services to the new ISDN networks and ultra-high-speed (100-Mbit-per-second) LANs and MANs. The tariff structure adopted by PTTs will be critical in determining both the rate of acceptance of ISDN and the way in which OSI standards are overlaid onto ISDN services. If ISDN circuits are inexpensive to set up and tear down, there should be a migration of OSI away from use of packet-switched toward circuit-switched networks. On the other hand, if the ISDN circuits are expensive, there could be considerable user resistance to the move away from current switched data networks and leased lines.

In either case, the adoption of OSI will accelerate rapidly through the 1990s. Existing standards and existing network architectures will form the basis of this growth initially, but later new LAN and ISDN architectures are likely to become increasingly important.

Evolution of the OSI Model

High-speed networks will give rise to major changes in the OSI model. New applications such as interactive graphics will need to make full use of the bandwidth available, and the structure of the current seven layers does not as yet make optimal use of the network. In addition, the functions of some of the layers are implicit in the application and impose an unnecessary overhead on the communication.

Another approach that may be appropriate for very high-performance applications is one where unnecessary layers remain inactive and the seven layers are collapsed into three or four simplified layers. This is analogous to the RISC (reduced instruction set computer) approach to processor design, which has been used to develop very high-performance processors by reducing the complexity of the instruction set.

However, changes to the concepts and structure of the OSI model will not appear quickly and can only be built on a firm basis of support of the existing model in the 1990s. The investment that manufacturers, users, and governments have placed in existing standards must reap significant returns before such radical changes will be considered.

Planning for the Future

For the user or provider of communications services, it is important to be aware of trends when planning for the future. Through the development of standards such as OSI, technologies are moving toward international standards.

In the messaging area, the acceptance of the MHS standards and growth of MHS products and services have been phenomenal in the past few years. For the movement of bulk data, the demand for FTAM products is growing rapidly, and for the transmission of industry-specific transactions, EDI networks and standards are growing rapidly. These trends will continue into the 1990s and will become increasingly important as OSI becomes a major force that competes with SNA.

It is likely that SNA will still continue to be important for some years to come and is ahead of OSI in some areas such as network management. OSI acceptance has been delayed in the past due to a number of weaknesses, notably the time that standards take to be agreed upon. These weaknesses are being addressed; for example, new procedures have been set in place to accelerate the progress of standards.

In addition, new technologies and applications will appear that will take time to be included in the standardization process. In these cases, proprietary solutions will be used. As often happens when a good proprietary solution has been found, it forms the basis of an eventual international standard. This is aided by the fact that many manufacturers develop new products within an OSI framework even when specific standards are not available.

In the long term, OSI will evolve and change to remove inefficiencies and accommodate new technologies. If these changes can be made quickly and incorporated into the model, OSI will become the dominant architecture of the 1990s and displace SNA, which has been the leading architecture during the 1980s.

*The adoption of OSI
will accelerate rapidly
through the 1990s.*

OSI Acronyms

AD	Addendum
ANSI	American National Standards Institute
AP	Application process
ACSE	Association control service element
CCITT	International Telegraph and Telephone Consultative Committee
CCR	Commitment, Concurrency, and Recovery
CEN/CENELEC	European Committee for Standardization/European Committee for Electrotechnical Standardization
CGMIF	Computer graphics metafile interchange format
CMIS	Common Management Information Services
CMIP	Common Management Information Protocols
CSMA/CD	Collision sense multiple access with collision detect
DAD	Draft Addendum
DIS	Draft International Standard
DP	Draft Proposal
EDI	Electronic Data Interchange
FTAM	File transfer, access, and management standards
GKS	Graphics kernel system
GOSIP	Government OSI profiles
HDLC	High level data link control
IEEE	Institute of Electrical and Electronic Engineers
IS	International Standard
ISO	International Organization for Standardization
IWU	Interworking unit
LAN	Local area network
LLC	Logical Link Control
LU	Logical unit
ODIF	Office document interchange format
MAN	Metropolitan area network
MAP	Manufacturing Automation Protocol
MHS	Message Handling System
MIB	Management information base
MOTIS	Message-Oriented Text Interchange System
ODA	Office Document Architecture
OSI	Open Systems Interconnection
OSIE	Open Systems Interconnection Environment
PDIF	Product definition interchange format
PTT	Postal, telephone, and telegraph service
RISC	Reduced instruction set computer
RJE	Remote job entry
SMI	Structure of Management Information
SMIS	Specific Management Information Services
SNA	Systems Network Architecture
TOP	Technical and Office Protocols
TP	Transaction processing
VT	Virtual Terminal
WAN	Wide area network

Conclusion

OSI was designed by international committees to provide interworking between heterogeneous computer systems. A complete set of protocols is now in place for several application areas, and work is proceeding to extend the range of application areas covered. The reference model is undergoing continual enhancement to incorporate new technologies, such as high-speed LANs, and new requirements, such as security and management.

OSI has the potential to offer enormous benefits to industry worldwide. With OSI, users will be able to install the equipment that is most suitable to the task, regardless of vendor. They will also be able to take advantage of data services offered anywhere on the globe and communicate directly with other independent companies.

To get maximum benefit from these opportunities, users will require products from vendors that fully integrate OSI functionality into their operating systems rather than select a subset of standards that can most easily be integrated with proprietary architectures. The number and range of products that implement OSI standards is already growing rapidly and will continue to increase in the 1990s as the move toward OSI networks gathers pace. OSI will move from being a minority interest to a major force and a mandatory requirement within corporate networks and for all business communications.

New services and new technologies will appear and place new demands on communication services. The OSI model will evolve to accommodate these changes by delivering new and enhanced capabilities in a timely manner.

References

- Open Systems Communication*. 1986. Omnicom, Inc.
- Open Systems Data Transfer*. 1986. Omnicom, Inc.
- CCITT Red Book Recommendations X.200-X.250*. Geneva, 1985.

Andrew Dunn joined Tandem in 1985. Since then he has worked in the European Consultancy Group providing technical and marketing support in the areas of communications and networking to the European Division. Prior to joining Tandem, he worked for a government research agency in the U.K., where he spent four years developing application software, six years developing networking and system software, and four years managing a distributed UNIX project.

The *Tandem Journal* became the *Tandem Systems Review* in February 1985. Four issues of the *Tandem Journal* were published:¹

Volume 1, Number 1	Fall 1983	Part no. 83930*
Volume 2, Number 1	Winter 1984	Part no. 83931*
Volume 2, Number 2	Spring 1984	Part no. 83932*
Volume 2, Number 3	Summer 1984	Part no. 83933*

As of this issue, 11 issues of the *Tandem Systems Review* have been published:

Volume 1, Number 1	February 1985	Part no. 83934*
Volume 1, Number 2	June 1985	Part no. 83935*
Volume 2, Number 1	February 1986	Part no. 83936*
Volume 2, Number 2	June 1986	Part no. 83937
Volume 2, Number 3	December 1986	Part no. 83938
Volume 3, Number 1	March 1987	Part no. 83939
Volume 3, Number 2	August 1987	Part no. 83940
Volume 4, Number 1	February 1988	Part no. 11078
Volume 4, Number 2	July 1988	Part no. 13693**
Volume 4, Number 3	October 1988	Part no. 15748
Volume 5, Number 1	April 1989	Part no. 18662

The articles published in all 15 issues are arranged by subject below. (*Tandem Journal* is abbreviated as TJ and *Tandem Systems Review* as TSR.) A second index, arranged by product, is also provided.

Index by Subject

Article title	Author(s)	Publication	Volume, Issue	Season or month and year	Part number
Application Development and Languages					
Ada: Tandem's Newest Compiler and Programming Environment	R. Vnuk	TSR	3,2	Aug. 1987	83940
An Introduction to Tandem EXTENDED BASIC*	J. Meyerson	TJ	2,2	Spring 1984	83932
State-of-the-Art C Compiler	E. Kit	TSR	2,2	June 1986	83937
Tandem's New COBOL85*	D. Nelson	TSR	2,1	Feb. 1986	83936
Debugging TACL Code**	L. Palmer	TSR	4,2	July 1988	13693
The ENABLE Program Generator for Multifile Applications*	B. Chapman, J. Zimmerman	TSR	1,1	Feb. 1985	83934
PATHFINDER—An Aid for Application Development*	S. Bennett	TJ	1,1	Fall 1983	83930
A New Design for the PATHWAY TCP*	R. Wong	TJ	2,2	Spring 1984	83932
PATHWAY IDS: A Message-level Interface to Devices and Processes	M. Anderton, M. Noonan	TSR	2,2	June 1986	83937
TACL, Tandem's New Extensible Command Language*	J. Campbell, R. Glascock	TSR	2,1	Feb. 1986	83936
New TAL Features	C. Lu, J. Murayama	TSR	2,2	June 1986	83837
TMF and the Multi-Threaded Requester*	T. Lemberger	TJ	1,1	Fall 1983	83930
Writing a Command Interpreter*	D. Wong	TSR	1,2	June 1985	83935
Customer Support					
Customer Information Service	J. Massucco	TSR	3,1	March 1987	83939
Remote Support Strategy	J. Eddy	TSR	3,1	March 1987	83939
Tandem's Software Support Plan	R. Baker, D. McEvoy	TSR	3,1	March 1987	83939

¹Articles and issues indicated by an asterisk (*) are no longer in stock. Issues indicated by a double asterisk (**) are available in limited quantities.

Article title	Author(s)	Publication	Volume, Issue	Season or month and year	Part number
Data Communications					
Changes in FOX*	N. Donde	TSR	1,2	June 1985	83935
Introduction to MULTILAN	A. Coyle	TSR	4,1	Feb. 1988	11078
Overview of the MULTILAN Server	A. Rowe	TSR	4,1	Feb. 1988	11078
A SNAX Passthrough Tutorial*	D. Kirk	TJ	2,2	Spring 1984	83932
SNAX/APC: Tandem's New SNA Software for Distributed Processing	B. Grantham	TSR	3,1	March 1987	83939
SNAX/HLS: An Overview*	S. Saltwick	TSR	1,2	June 1985	83935
Using the MULTILAN Application Interfaces	M. Berg, A. Rowe	TSR	4,1	Feb. 1988	11078
Data Management					
A Comparison of the B00 DP1 and DP2 Disc Processes*	T. Schachter	TSR	1,2	June 1985	83935
DP1-DP2 File Conversion: An Overview*	J. Tate	TSR	2,1	Feb. 1986	83936
DP2's Efficient Use of Cache*	T. Schachter	TSR	1,2	June 1985	83935
DP2 Highlights*	K. Carlyle, L. McGowan	TSR	1,2	June 1985	83935
DP2 Key-sequenced Files*	T. Schachter	TSR	1,2	June 1985	83935
Determining FCP Conversion Time*	J. Tate	TSR	2,1	Feb. 1986	83936
Improvements in TMF*	T. Lemberger	TSR	1,2	June 1985	83935
High-Performance SQL Through Low-Level System Integration**	A. Borr	TSR	4,2	July 1988	13693
Overview of NonStop SQL**	H. Cohen	TSR	4,2	July 1988	13693
NetBatch: Managing Batch Processing on Tandem Systems	D. Wakashige	TSR	5,1	April 1989	18662
NonStop SQL Data Dictionary**	R. Holbrook, D. Tsou	TSR	4,2	July 1988	13693
NonStop SQL Optimizer: Basic Concepts**	M. Pong	TSR	4,2	July 1988	13693
NonStop SQL Optimizer: Query Optimization and User Influence**	M. Pong	TSR	4,2	July 1988	13693
NonStop SQL Reliability**	C. Fenner	TSR	4,2	July 1988	13693
The Relational Data Base Management Solution*	G. Ow	TJ	2,1	Winter 1984	83931
Tandem's NonStop SQL Benchmark	Tandem Performance Group	TSR	4,1	Feb. 1988	11078
TMF Autorollback: A New Recovery Feature*	M. Pong	TSR	1,1	Feb. 1985	83934
The TRANSFER Delivery System for Distributed Applications*	S. Van Pelt	TJ	2,2	Spring 1984	83932
Manuals/Courses					
B00 Software Manuals*	S. Olds	TSR	1,2	June 1985	83935
C00 Software Manuals	E. Levi	TSR	4,1	Feb. 1988	11078
New Software Courses*	M. Janow	TSR	1,2	June 1985	83935
New Software Courses	J. Limper	TSR	4,1	Feb. 1988	11078
Subscription Policy for Software Manuals*	T. McSweeney	TSR	2,1	Feb. 1986	83936
Tandem's New Products*	C. Robinson	TSR	2,1	Feb. 1986	83936
Tandem's New Products	C. Robinson	TSR	2,2	June 1986	83937
Operating Systems					
The GUARDIAN Message System and How to Design for It*	M. Chandra	TSR	1,1	Feb. 1985	83935
Highlights of the B00 Software Release*	K. Coughlin, R. Montevaldo	TSR	1,2	June 1985	83935
Overview of the C00 Release	L. Marks	TSR	4,1	Feb. 1988	11078
Increased Code Space*	A. Jordan	TSR	1,2	June 1985	83935
Managing System Time Under GUARDIAN 90*	E. Nellen	TSR	2,1	Feb. 1986	83936
New GUARDIAN 90 Timekeeping Facilities*	E. Nellen	TSR	1,2	June 1985	83935
New Process-timing Features*	S. Sharma	TSR	1,2	June 1985	83935
NonStop II Memory Organization and Extended Addressing*	D. Thomas	TJ	1,1	Fall 1983	83930
Robustness to Crash in a Distributed Data Base: A Nonshared-memory Approach*	A. Borr	TSR	1,2	June 1985	83935
The Tandem Global Update Protocol*	R. Carr	TSR	1,2	June 1985	83935

Article title	Author(s)	Publication	Volume, Issue	Season or month and year	Part number
Performance and Capacity Planning					
Tandem 5200 Optical Storage Facility: Performance and Optimization Considerations	S. Coleman	TSR	5,1	April 1989	18662
Buffering for Better Application Performance*	R. Mattran	TSR	2,1	Feb. 1986	83936
Capacity Planning Concepts	R. Evans	TSR	2,3	Dec. 1986	83938
C00 TMD5 Performance	J. Mead	TSR	4,1	Feb. 1988	11078
Estimating Host Response Time in a Tandem System	H. Horwitz	TSR	4,3	Oct. 1988	15748
A Performance Retrospective	P. Oleinick, P. Shah	TSR	2,3	Dec. 1986	83938
The Performance Characteristics of Tandem NonStop Systems*	J. Day	TJ	1,1	Fall 1983	83930
Performance Considerations for Application Processes	R. Glasstone	TSR	2,3	Dec. 1986	83938
Optimizing Sequential Processing on the Tandem System*	R. Welsh	TJ	2,3	Summer 1984	83933
Predicting Response Time in On-line Transaction Processing Systems	A. Khatri	TSR	2,2	June 1986	83937
The 6600 and TCC6820 Communications Controllers: A Performance Comparison	P. Beadles	TSR	2,3	Dec. 1986	83938
Improved Performance for BACKUP2 and RESTORE2*	A. Khatri, M. McCline	TSR	1,2	June 1985	83935
Credit-authorization Benchmark for High Performance and Linear Growth*	T. Chmiel, T. Houy	TSR	2,1	Feb. 1986	83936
DP2 Performance*	J. Enright	TSR	1,2	June 1985	83935
The ENCORE Stress Test Generator for On-line Transaction Processing Applications*	S. Kosinski	TJ	2,1	Winter 1984	83931
FASTSORT: An External Sort Using Parallel Processing	J. Gray, M. Stewart, A. Tsukerman, S. Uren, B. Vaughan	TSR	2,3	Dec. 1986	83938
Performance Measurements of an ATM Network Application	N. Cabell, D. Mackie	TSR	2,3	Dec. 1986	83938
How to Set Up a Performance Data Base with MEASURE and ENFORM	M. King	TSR	2,3	Dec. 1986	83938
MEASURE: Tandem's New Performance Measurement Tool	D. Dennison	TSR	2,3	Dec. 1986	83938
Message System Performance Enhancements	D. Kinkade	TSR	2,3	Dec. 1986	83938
Message System Performance Tests	S. Uren	TSR	2,3	Dec. 1986	83938
The PATHWAY TCP: Performance and Tuning*	J. Vatz	TSR	1,1	Feb. 1985	83934
Understanding PATHWAY Statistics*	M. Pong	TJ	2,2	Spring 1984	83932
Sizing Cache for Applications that Use B-series DP1 and TMF	P. Shah	TSR	2,2	June 1986	83937
Sizing the Spooler Collector Data File	H. Norman	TSR	4,1	Feb. 1988	11078
Tandem's Approach to Fault Tolerance	B. Ball, W. Bartlett, S. Thompson	TSR	4,1	Feb. 1988	11078
Getting Optimum Performance from Tandem Tape Systems	A. Khatri	TSR	2,3	Dec. 1986	83938
NonStop VLX Performance	J. Enright	TSR	2,3	Dec. 1986	83938
Peripherals					
5120 Tape Subsystem Recording Technology	W. Phillips	TSR	3,2	Aug. 1987	83940
Tandem 5200 Optical Storage Facility: A Hardware Perspective	A. Patel	TSR	5,1	April 1989	18662
The 6100 Communications Subsystem: A New Architecture*	R. Smith	TJ	2,1	Winter 1984	83931
The 6600 and TCC6820 Communications Controllers: A Performance Comparison	P. Beadles	TSR	2,3	Dec. 1986	83938
Data-Encoding Technology Used in the XL8 Storage Facility	D.S. Ng	TSR	2,2	June 1986	83937
Data-Window Phase-Margin Analysis	A. Painter, H. Pham, H. Thomas	TSR	2,2	June 1986	83937
The DYNAMITE Workstation: An Overview*	G. Smith	TSR	1,2	June 1985	83935
An Introduction to DYNAMITE Workstation Host Integration*	S. Kosinski	TSR	1,2	June 1985	83935
Introducing the 3207 Tape Controller*	S. Chandran	TSR	1,2	June 1985	83935
The Model 6VI Voice Input Option: Its Design and Implementation*	B. Huggett	TJ	2,3	Summer 1984	83933
The Role of Optical Storage in Information Processing	L. Sabaroff	TSR	3,2	Aug. 1987	83940
Peripheral Device Interfaces	J. Blakkan	TSR	3,2	Aug. 1987	83940
Streaming Tape Drives	J. Blakkan	TSR	3,2	Aug. 1987	83940
Plated Media Technology Used in the XL8 Storage Facility	D.S. Ng	TSR	2,2	June 1986	83937
The V8 Disc Storage Facility: Setting a New Standard for On-line Disc Storage*	M. Whiteman	TSR	1,2	June 1985	83935

Article title	Author(s)	Publication	Volume, Issue	Season or month and year	Part number
Processors					
The High-Performance NonStop TXP Processor*	W. Bartlett, T. Houy, D. Meyer	TJ	2,1	Winter 1984	83931
The NonStop TXP Processor: A Powerful Design for On-Line Transaction Processing*	P. Oleinick	TJ	2,3	Summer 1984	83933
NonStop CLX—Optimized for Distributed On-line Transaction Processing	D. Lenoski	TSR	5,1	April 1989	18662
NonStop VLX Hardware Design	M. Brown	TSR	2,3	Dec. 1986	83938
The VLX: A Design for Serviceability	J. Allen, R. Boyle	TSR	3,1	March 1987	83939
Security					
Distributed Protection with SAFEGUARD	T. Chou	TSR	2,2	June 1986	83937
System Connectivity					
The OSI Model: Overview, Status, and Current Issues	A. Dunn	TSR	5,1	April 1989	18662
Terminal Connection Alternatives for Tandem Systems	J. Simonds	TSR	5,1	April 1989	18662
System Management					
Configuring Tandem Disk Subsystems	S. Sittler	TSR	2,3	Dec. 1986	83938
Data Replication in Tandem's Distributed Name Service	T. Eastep	TSR	4,3	Oct. 1988	15748
Enhancements to TMDS	L. White	TSR	3,2	Aug. 1987	83940
Event Management Service Design and Implementation	H. Jordan, R. McKee, R. Schuet	TSR	4,3	Oct. 1988	15748
Introducing TMDS, Tandem's New On-line Diagnostic System*	J. Troisi	TSR	1,2	June 1985	83935
Overview of DSM	P. Homan, B. Malizia, E. Reisner	TSR	4,3	Oct. 1988	15748
Network Statistics System	M. Miller	TSR	4,3	Oct. 1988	15748
SCP and SCF: A General Purpose Implementation of the Subsystem Programmatic Interface	T. Lawson	TSR	4,3	Oct. 1988	15748
Tandem's Subsystem Programmatic Interface	G. Tom	TSR	4,3	Oct. 1988	15748
Using FOX to Move a Fault-tolerant Application*	C. Breighner	TSR	1,1	Feb. 1985	83934
Using the Subsystem Programmatic Interface and Event Management Services	K. Stobie	TSR	4,3	Oct. 1988	15748
VIEWPOINT Operations Console Facility	R. Hansen, G. Stewart	TSR	4,3	Oct. 1988	15748
VIEWSYS: An On-line System-resource Monitor*	D. Montgomery	TSR	1,2	June 1985	83935
Utilities					
Enhancements to PS MAIL	R. Funk	TSR	3,1	March 1987	83939

Index by Project

Article title	Author(s)	Publication	Volume, Issue	Season or month and year	Part number
3207 Tape Controller					
Introducing the 3207 Tape Controller*	S. Chandran	TSR	1,2	June 1985	83935
5120 Tape Subsystem					
5120 Tape Subsystem Recording Technology	W. Phillips	TSR	3,2	Aug. 1987	83940
5200 Optical Storage					
Tandem 5200 Optical Storage Facility: A Hardware Perspective	A. Patel	TSR	5,1	April 1989	18662
Tandem 5200 Optical Storage Facility: Performance and Optimization Considerations	S. Coleman	TSR	5,1	April 1989	18662
The Role of Optical Storage in Information Processing	L. Sabaroff	TSR	4,1	Feb. 1988	11078
6100 Communications Subsystem					
The 6100 Communications Subsystem: A New Architecture*	R. Smith	TJ	2,1	Winter 1984	83931
6530 Terminal					
The Model 6VI Voice Input Option: Its Design and Implementation*	B. Huggett	TJ	2,3	Summer 1984	83933
6600 and TCC6820 Communications Controllers					
The 6600 and TCC6820 Communications Controllers: A Performance Comparison	P. Beadles	TSR	2,3	Dec. 1986	83938
ADA					
Ada: Tandem's Newest Compiler and Programming Environment	R. Vnuk	TSR	3,2	Aug. 1987	83940
BASIC					
An Introduction to Tandem EXTENDED BASIC*	J. Meyerson	TJ	2,2	Spring 1984	83932
COMINT (CI)					
State-of-the-art C Compiler	E. Kit	TSR	2,2	June 1986	83937
CI					
Writing a Command Interpreter*	D. Wong	TSR	1,2	June 1985	83935
CIS					
Customer Information Service	J. Massucco	TSR	3,1	March 1987	83939
COBOL85					
Tandem's New COBOL85*	D. Nelson	TSR	2,1	Feb. 1986	83936
CLX					
NonStop CLX—Optimized for Distributed On-line Transaction Processing	D. Lenoski	TSR	5,1	April 1989	18662
DP1 and DP2					
Sizing Cache for Applications that Use B-series DP1 and TMF	P. Shah	TSR	2,2	June 1986	83937
A Comparison of the B00 DP1 and DP2 Disc Processes*	T. Schachter	TSR	1,2	June 1985	83935
P1-DP2 File Conversion: An Overview*	J. Tate	TSR	2,1	Feb. 1986	83936
DP2's Efficient Use of Cache*	T. Schachter	TSR	1,2	June 1985	83935
DP2 Highlights*	K. Carlyle, L. McGowan	TSR	1,2	June 1985	83935
DP2 Key-sequenced Files*	T. Schachter	TSR	1,2	June 1985	83935
DP2 Performance*	J. Enright	TSR	1,2	June 1985	83935
Determining FCP Conversion Time*	J. Tate	TSR	2,1	Feb. 1986	83936
DSM					
Data Replication in Tandem's Distributed Name Service	T. Eastep	TSR	4,3	Oct. 1988	15748
Event Management Service Design and Implementation	H. Jordan, R. McKee, R. Schuet	TSR	4,3	Oct. 1988	15748
Overview of DSM	P. Homan, B. Malizia, E. Reisner	TSR	4,3	Oct. 1988	15748
Network Statistics System	M. Miller	TSR	4,3	Oct. 1988	15748
SCP and SCF: A General Purpose Implementation of the Subsystem Programmatic Interface	T. Lawson	TSR	4,3	Oct. 1988	15748
Tandem's Subsystem Programmatic Interface	G. Tom	TSR	4,3	Oct. 1988	15748
Using the Subsystem Programmatic Interface and Event Management Services	K. Stobie	TSR	4,3	Oct. 1988	15748
VIEWPOINT Operations Console Facility	R. Hansen, G. Stewart	TSR	4,3	Oct. 1988	15748

Article title	Author(s)	Publication	Volume, Issue	Season or month and year	Part number
DYNAMITE					
The DYNAMITE Workstation: An Overview*	G. Smith	TSR	1,2	June 1985	83935
An Introduction to DYNAMITE Workstation Host Integration*	S. Kosinski	TSR	1,2	June 1985	83935
ENABLE					
The ENABLE Program Generator for Multifile Applications*	B. Chapman, J. Zimmerman	TSR	1,1	Feb. 1985	83934
ENCOMPASS					
The Relational Data Base Management Solution*	G. Ow	TJ	2,1	Winter 1984	83931
ENCORE					
The ENCORE Stress Test Generator for On-line Transaction Processing Applications*	S. Kosinski	TJ	2,1	Winter 1984	83931
FASTSORT					
FASTSORT: An External Sort Using Parallel Processing	J. Gray, M. Stewart, A. Tsukerman, S. Uren, B. Vaughan	TSR	2,3	Dec. 1986	83938
FOX					
Changes in FOX*	N. Donde	TSR	1,2	June 1985	83935
Using FOX to Move a Fault-tolerant Application*	C. Breighner	TSR	1,1	Feb. 1985	83934
GUARDIAN 90					
Highlights of the B00 Software Release*	K. Coughlin, R. Montevaldo	TSR	1,2	June 1985	83935
B00 Software Manuals*	S. Olds	TSR	1,2	June 1985	83935
C00 TMDS Performance	J. Meade	TSR	4,1	Feb. 1988	11078
C00 Software Manuals	E. Levi	TSR	4,1	Feb. 1988	11078
Overview of the C00 Release	L. Marks	TSR	4,1	Feb. 1988	11078
The GUARDIAN Message System and How to Design for It*	M. Chandra	TSR	1,1	Feb. 1985	83935
Improved Performance for BACKUP2 and RESTORE2*	A. Khatri, M. McCline	TSR	1,2	June 1985	83935
Increased Code Space*	A. Jordan	TSR	1,2	June 1985	83935
Introducing TMDS, Tandem's New On-line Diagnostic System*	J. Troisi	TSR	1,2	June 1985	83935
Enhancements to TMDS	L. White	TSR	3,2	Aug. 1987	83940
Managing System Time Under GUARDIAN 90*	E. Nellen	TSR	2,1	Feb. 1986	83936
Message System Performance Enhancements	D. Kinkade	TSR	2,3	Dec. 1986	83938
Message System Performance Tests	S. Uren	TSR	2,3	Dec. 1986	83938
New GUARDIAN 90 Timekeeping Facilities*	E. Nellen	TSR	1,2	June 1985	83935
New Process-timing Features*	S. Sharma	TSR	1,2	June 1985	83935
NonStop II Memory Organization and Extended Addressing*	D. Thomas	TJ	1,1	Fall 1983	83930
Robustness to Crash in a Distributed Data Base: A Nonshared-memory Multiprocessor Approach*	A. Borr	TSR	1,2	June 1985	83935
The Tandem Global Update Protocol*	R. Carr	TSR	1,2	June 1985	83935
Tandem's Approach to Fault Tolerance	B. Ball, W. Bartlett, S. Thompson	TSR	4,1	Feb. 1988	11078
MEASURE					
How to Set Up a Performance Data Base with MEASURE and ENFORM	M. King	TSR	2,3	Dec. 1986	83938
MEASURE: Tandem's New Performance Measurement Tool	D. Dennison	TSR	2,3	Dec. 1986	83938
MULTILAN					
Introduction to MULTILAN	A. Coyle	TSR	4,1	Feb. 1988	11078
Overview of the MULTILAN Server	A. Rowe	TSR	4,1	Feb. 1988	11078
Using the MULTILAN Application Interfaces	M. Berg, A. Rowe	TSR	4,1	Feb. 1988	11078
NetBatch					
NetBatch: Managing Batch Processing on Tandem Systems	D. Wakashige	TSR	5,1	April 1989	18662
OSI					
The OSI Model: Overview, Status, and Current Issues	A. Dunn	TSR	5,1	April 1989	18662

Article title	Author(s)	Publication	Volume, Issue	Season or month and year	Part number
PATHFINDER					
PATHFINDER—An Aid for Application Development*	S. Benett	TJ	1,1	Fall 1983	83930
PATHWAY					
PATHWAY IDS: A Message-level Interface to Devices and Processes	M. Anderton, M. Noonan	TSR	2,2	June 1986	83937
A New Design for the PATHWAY TCP*	R. Wong	TJ	2,2	Spring 1984	83932
The PATHWAY TCP: Performance and Tuning	J. Vatz	TSR	1,1	Feb. 1985	83934
Understanding PATHWAY Statistics*	M. Pong	TJ	2,2	Spring 1984	83932
PS MAIL					
Enhancements to PS MAIL	R. Funk	TSR	3,1	March 1987	83939
SAFEGUARD					
Distributed Protection with SAFEGUARD	T. Chou	TSR	2,2	June 1986	83937
SNAX					
A SNAX Passthrough Tutorial*	D. Kirk	TJ	2,2	Spring 1984	83932
SNAX/APC: Tandem's New SNA Software for Distributed Processing	B. Grantham	TSR	3,1	March 1987	83939
SNAX/HLS: An Overview*	S. Saltwick	TSR	1,2	June 1985	83935
SPOOLER					
Sizing the Spooler Collector Data File	H. Norman	TSR	4,1	Feb. 1988	11078
SQL					
High-Performance SQL Through Low-Level System Integration**	A. Borr	TSR	4,2	July 1988	13693
NonStop SQL Data Dictionary**	R. Holbrook, D. Tsou	TSR	4,2	July 1988	13693
NonStop SQL Optimizer: Basic Concepts**	M. Pong	TSR	4,2	July 1988	13693
NonStop SQL Optimizer: Query Optimization and User Influence**	M. Pong	TSR	4,2	July 1988	13693
NonStop SQL Reliability**	C. Fenner	TSR	4,2	July 1988	13693
Overview of NonStop SQL**	H. Cohen	TSR	4,2	July 1988	13693
Tandem's Nonstop SQL Benchmark	Tandem Performance Group	TSR	4,1	Feb. 1988	11078
TACL					
Debugging TACL Code**	L. Palmer	TSR	4,2	July 1988	13693
TACL, Tandem's New Extensible Command Language*	J. Campbell, R. Glascock	TSR	2,1	Feb. 1986	83936
TAL					
New TAL Features	C. Lu, J. Murayama	TSR	2,2	June 1986	83837
TMF					
Improvements in TMF*	T. Lemberger	TSR	1,2	June 1985	83935
TMF and the Multi-Threaded Requester*	T. Lemberger	TJ	1,1	Fall 1983	83930
TMF Autorollback: A New Recovery Feature*	M. Pong	TSR	1,1	Feb. 1985	83934
TRANSFER					
The TRANSFER Delivery System for Distributed Applications*	S. Van Pelt	TJ	2,2	Spring 1984	83932
TXP					
The High-Performance NonStop TXP Processor*	W. Bartlett, T. Houy, D. Meyer	TJ	2,1	Winter 1984	83931
The NonStop TXP Processor: A Powerful Design for On-line Transaction Processing*	P. Oleinick	TJ	2,3	Summer 1984	83933
V8					
The V8 Disc Storage Facility: Setting a New Standard for On-line Disc Storage*	M. Whiteman	TSR	1,2	June 1985	83935
VIEWSYS					
VIEWSYS: An On-line System-resource Monitor*	D. Montgomery	TSR	1,2	June 1985	83935
VLX					
VLX Hardware Design	M. Brown	TSR	2,3	Dec. 1986	83938
NonStop VLX Performance	J. Enright	TSR	2,3	Dec. 1986	83938
The VLX: A Design for Serviceability	J. Allen, R. Boyle	TSR	3,1	March 1987	83939

Article title	Author(s)	Publication	Volume, Issue	Season or month and year	Part number
XL8					
Data-encoding Technology Used in the XL8 Storage Facility	D.S. Ng	TSR	2,2	June 1986	83937
Plated Media Technology Used in the XL8 Storage Facility	D.S. Ng	TSR	2,2	June 1986	83937
Miscellaneous²					
Performance Measurements of an ATM Network Application	N. Cabell, D. Mackie	TSR	2,3	Dec. 1986	83938
Buffering for Better Application Performance*	R. Mattran	TSR	2,1	Feb. 1986	83936
Capacity Planning Concepts	R. Evans	TSR	2,3	Dec. 1986	83938
Credit-authorization Benchmark for High Performance and Linear Growth*	T. Chmiel, T. Houy	TSR	2,1	Feb. 1986	83936
Data-window Phase-margin Analysis	A. Painter, H. Pham, H. Thomas	TSR	2,2	June 1986	83937
Configuring Tandem Disk Subsystems	S. Stiler	TSR	2,3	Dec. 1986	83938
Estimating Host Response Time in a Tandem System	H. Horwitz	TSR	4,3	Oct. 1988	15748
The Role of Optical Storage in Information Processing	L. Sabaroff	TSR	3,2	Aug. 1987	83940
Peripheral Device Interfaces	J. Blakkan	TSR	3,2	Aug. 1987	83940
A Performance Retrospective	P. Oleinick	TSR	2,3	Dec. 1986	83938
Performance Considerations for Application Processes	R. Glasstone	TSR	2,3	Dec. 1986	83938
The Performance Characteristics of Tandem NonStop Systems*	J. Day	TJ	1,1	Fall 1983	83930
Predicting Response Time in On-line Transaction Processing Systems	A. Khatri	TSR	2,2	June 1986	83937
Remote Support Strategy	J. Eddy	TSR	3,1	March 1987	83939
Optimizing Sequential Processing on the Tandem System*	R. Welsh	TJ	2,3	Summer 1984	83933
Tandem's Software Support Plan	R. Baker, D. McEvoy	TSR	3,1	March 1987	83939
Streaming Tape Drives	J. Blakkan	TSR	3,2	Aug. 1987	83940
Terminal Connection Alternatives for Tandem Systems	J. Simonds	TSR	5,1	April 1989	18662
Getting Optimum Performance from Tandem Tape Systems	A. Khatri	TSR	2,3	Dec. 1986	83938
New Software Courses*	M. Janow	TSR	1,2	June 1985	83935
New Software Courses*	J. Limper	TSR	4,1	Feb. 1988	11078
Subscription Policy for Software Manuals*	T. McSweeney	TSR	2,1	Feb. 1986	83936
Tandem's New Products*	C. Robinson	TSR	2,1	Feb. 1986	83936
Tandem's New Products	C. Robinson	TSR	2,2	June 1986	83937

²This category contains *Tandem Systems Review* articles that contain product information, but are not specifically product-related.

T A N D E M P U B L I C A T I O N S O R D E R F O R M

Subscriptions to the *Tandem Systems Review* are free. Use this form to subscribe, change a subscription, and order back copies.

For requests within the U.S., send this form to:

Tandem Computers Incorporated
Tandem Systems Review
18922 Forge Drive, LOC 216-05
Cupertino, CA 95014-0701

For requests *outside* the U.S., send this form to your local Tandem sales office.

Check the appropriate box(es):

- New subscription (# of copies desired _____)
- Subscription change (# of copies desired _____)
- Request for back copies. (Shipment subject to availability.)

To order back copies, write the number of copies next to the title(s) below. Please allow six to eight weeks for delivery.

NUMBER OF COPIES	<i>Tandem Systems Review</i>
_____	Part No. 83937, Vol. 2, No. 2, June 1986
_____	Part No. 83938, Vol. 2, No. 3, December 1986
_____	Part No. 83939, Vol. 3, No. 1, March 1987
_____	Part No. 83940, Vol. 3, No. 2, August 1987
_____	Part No. 11078, Vol. 4, No. 1, February 1988
_____	Part No. 13693, Vol. 4, No. 2, July 1988
_____	Part No. 15748, Vol. 4, No. 3, October 1988
_____	Part No. 18662, Vol. 5, No. 1, April 1989

Print your current address here:

COMPANY NAME

ADDRESS

ATTENTION

PHONE NUMBER (U.S.)

If your address has changed, print the old one here:

COMPANY NAME

ADDRESS

ATTENTION

PHONE NUMBER (U.S.)

MARC BRANDIFINO
DEPT 8449
LOC NUM 50-00
NEW YORK NY DISTRICT

Jandem Computers Incorporated
19333 Valico Parkway
Cupertino, CA 95014-2599