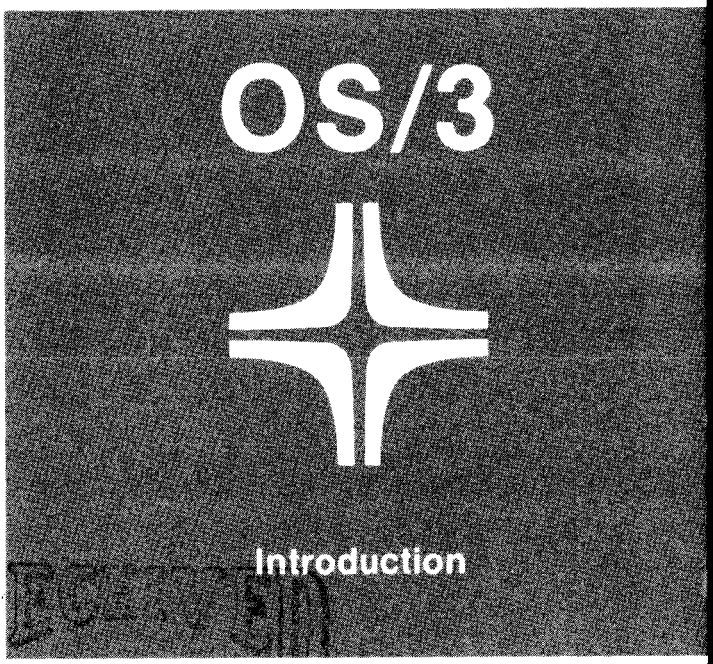
Integrated Communications Access Method (ICAM)



SPERRY

UP-8846 Rev. 1

This document contains the latest information available at the time of preparation. Therefore, it may contain descriptions of functions not implemented at manual distribution time. To ensure that you have the latest information regarding levels of implementation and functional availability, please consult the appropriate release documentation or contact your local Sperry representative.

Sperry reserves the right to modify or revise the content of this document. No contractual obligation by Sperry regarding level, scope, or timing of functional implementation is either expressed or implied in this document. It is further understood that in consideration of the receipt or purchase of this document, the recipient or purchaser agrees not to reproduce or copy it by any means whatsoever, nor to permit such action by others, for any purpose without prior written permission from Sperry.

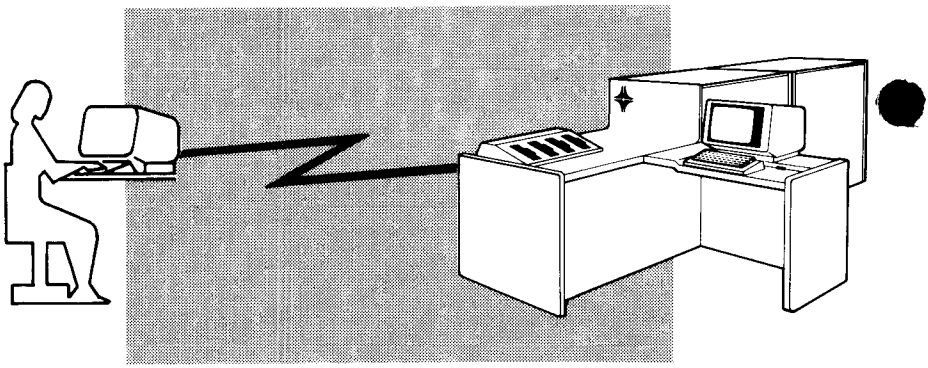
This document was prepared by Systems Publications using the SPERRY UTS 400 Text Editor. It was printed and distributed by the Customer Information Distribution Center (CIDC), 555 Henderson Rd., King of Prussia, Pa., 19406.

ICAM – simplifies communications

In the early days of computers, when you wanted a job done, you took it to the computer. Maybe you'd get the results the same day, but it might take a couple of days.

This was better than no computer but the user wanted faster results and he wanted the computer to come to him. Thus, terminals and hardware were developed that now allow you to *talk* to your computer from your workplace over telephone and telegraph wires.

Computer communications had arrived! For programmers, however, this meant complications. Now their programs had to address each outgoing message, encode it into a language understandable to a terminal, and constantly check for errors. It was like mailing a letter by taking it to the post office, sorting it, and making sure the right mailman got it.



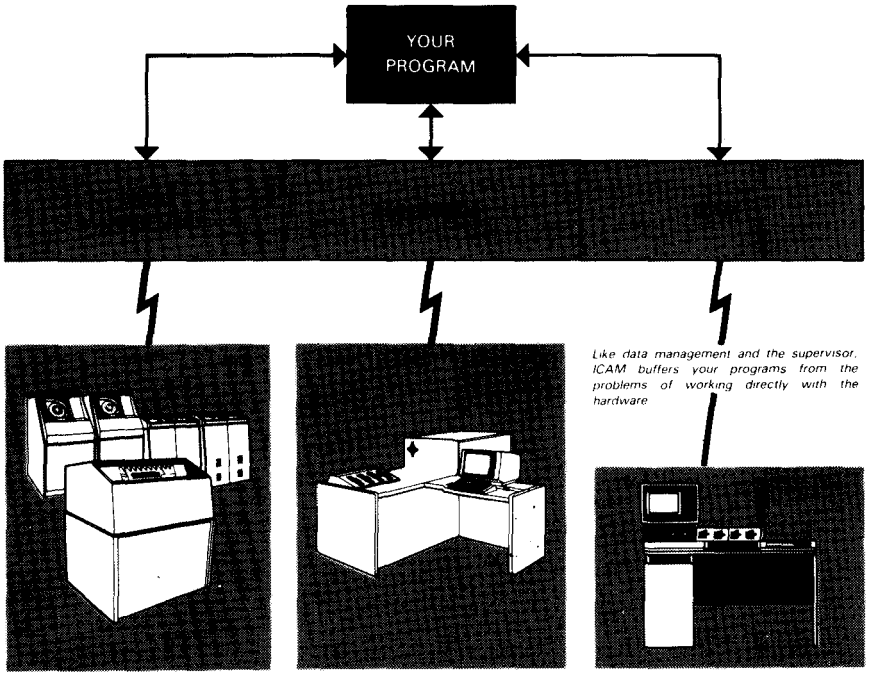
The Integrated Communications Access Method – ICAM – is Sperry's answer to these complications. ICAM does the hard work of addressing the message, encoding it, checking for errors, and a dozen other tasks your program used to do.

Now your program just processes the message contents. To send a message, your program hands it to ICAM; to receive a message, your program asks ICAM for it. Compared to the old way, this is like putting your message into a mailbox and forgetting it.

Because ICAM simplifies communications programming, applications that before were complex and/or costly are now practical. Your programmer can now spend more time writing business applications and less time on the mechanics of communications.

Using ICAM means that it's just as easy for your programmer to communicate with a terminal 1,000 miles away as it is to work with a file on a disk drive located next door. A program working with ICAM uses special instructions to read or write messages to and from terminals. These special instructions are the only real difference between a program working with ICAM and any other program working with files.

It's unnecessary for your programmer to know the internal workings of ICAM. To your programer, ICAM is invisible, just as the workings of the post office are invisible to you. To mail a letter, you only need to know where the mailbox is, not how the post office handles mail.



Like data management and the supervisor, ICAM buffers your programs from the problems of working directly with the hardware

LOCAL PERIPHERALS

CENTRAL PROCESSOR RESOURCES

REMOTE PERIPHERALS

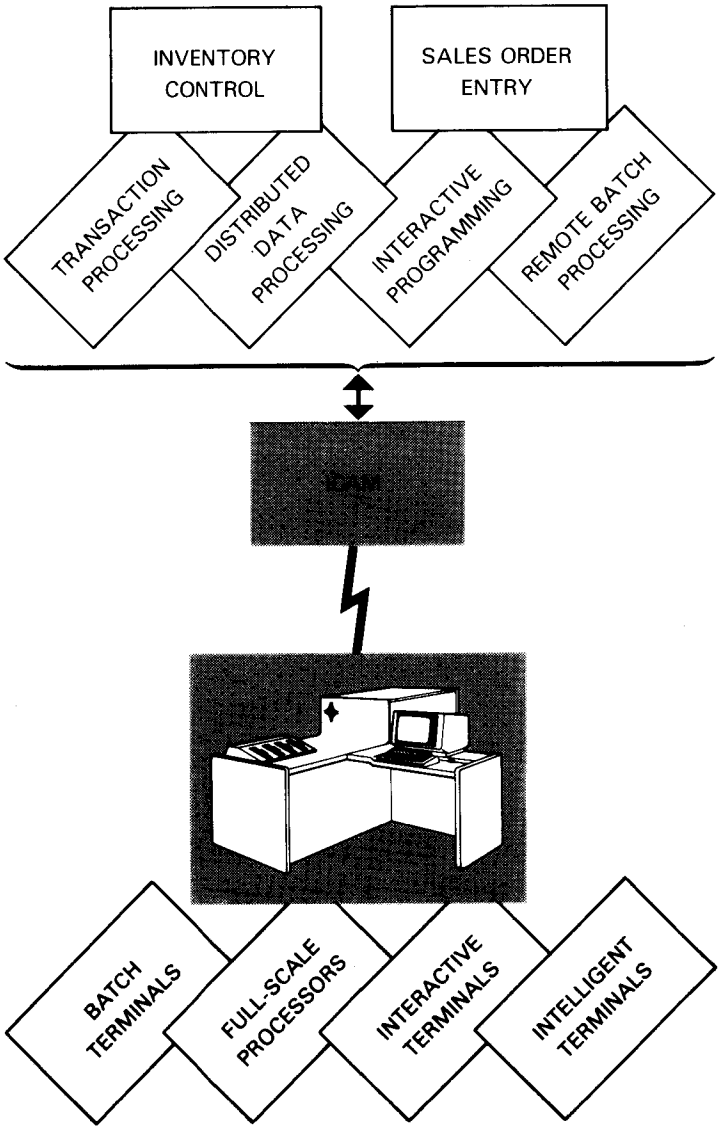
ICAM is flexible. You decide what you want, write the programs, purchase or rent the terminals, and then tailor ICAM to fit your communications needs. You aren't locked into a one-size-fits-all communications system. ICAM gives you what you need – nothing more, nothing less.

Because of this flexibility, your entire communications system is flexible. You pick the terminals that best suit your needs. They can be interactive, batch, intelligent, or full-scale processors – ICAM supports the terminals you choose.

Nor does ICAM limit the applications you can use. It works with transaction processing systems, remote batch processing systems, distributed data processing systems, and interactive (demand) systems.

With ICAM, your communications system is tailored to your business.

ICAM – Flexibility!



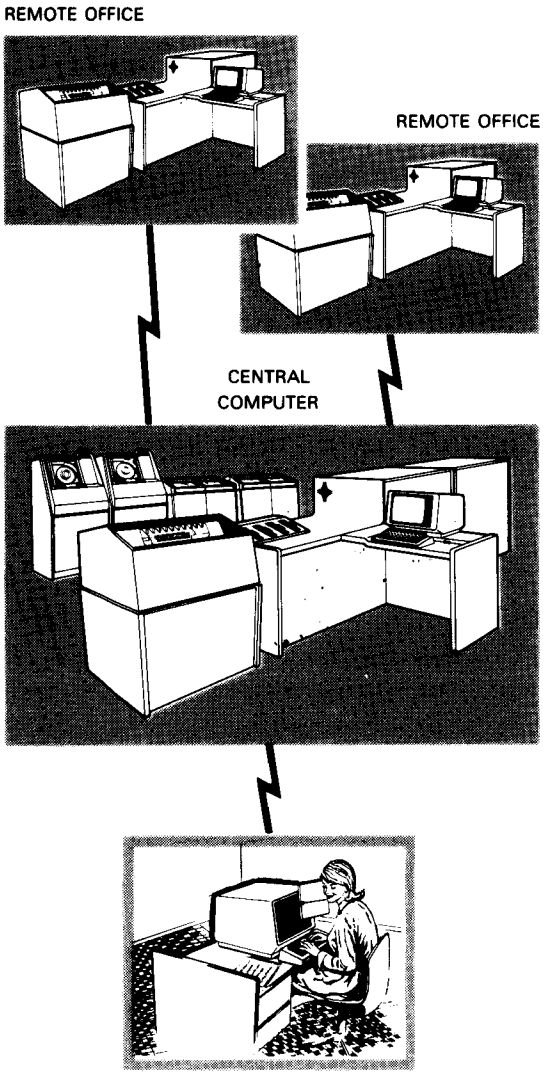
remote batch processing

The most common type of computer program is the batch job. The batch job does its processing, produces output, and then halts. Take a payroll program... It processes large amounts of data – time cards, salaries, deductions – and produces paychecks and updated payroll records. Once this payroll is completed, the program isn't used again until you need the next payroll.

If your operation has a number of locations, you want each of them to have the power of batch processing, but you don't want the cost of a separate computer for each of them. Sperry's remote batch processor is the answer.

Going back to the payroll example... You have six plants scattered across the country and every Wednesday each plant processes its own payroll so the checks will be ready Friday. With the remote batch processor, your central computer does each plant's payroll using data sent from each plant through ICAM. The remote batch processor then sends the paychecks and payroll records back to the individual plants.

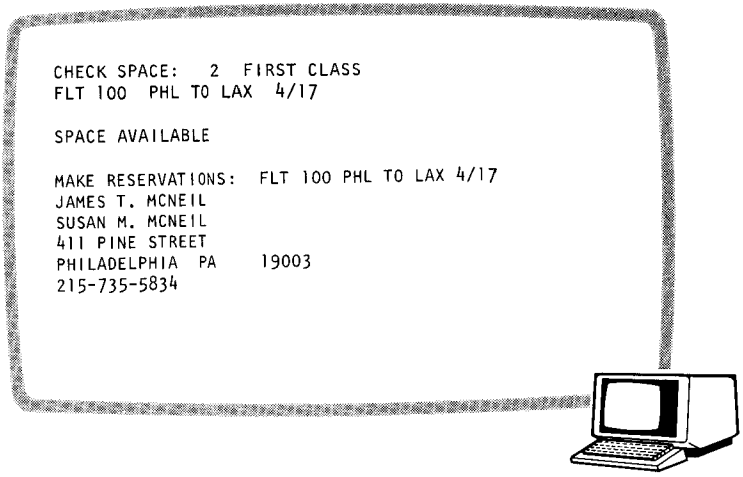
The remote batch processor is also a mailman. Let's say your Chicago sales office produces sales statistics for that office on a computer in Philadelphia. The Chicago office can send the statistics just to Chicago, to Chicago and Philadelphia, or to every office in the network.



transaction processing – information at your fingertips

You want to make two first-class air reservations – Philadelphia to Los Angeles for April 17. You call your airline and a few minutes later you have the reservations. It doesn't matter where you call from or whether you're leaving this afternoon or next month. With hundreds of flights and thousands of reservations, the airline can service you that quickly.

How? With the power of transaction processing. When you call, the reservations agent has the computer check the files for available seating accommodations. If seats are available, the reservations are made. Otherwise, the agent tries another flight. The agent doesn't know a thing about programming. He just makes inquiries from a terminal and the computer responds.



How does a transaction processing system work? A terminal operator sends an inquiry in to a program. Based on the instructions in the inquiry, the program either updates a data file or retrieves data from it. Then, it responds with a message to the terminal. The response can be an acknowledgment that the file has been updated, a single item of information, a list, or even a summary of all the data in the file.

When speed counts – when many people need the latest information – when paperwork costs become too high – transaction processing is the answer.

Following are three examples of how transaction processing can save time and money.

- Inventory Control System

  Too much or too little inventory costs you money. With transaction processing, you know how much you have – and how fast it's moving. When stock runs low, your transaction processing system prints a reorder slip.

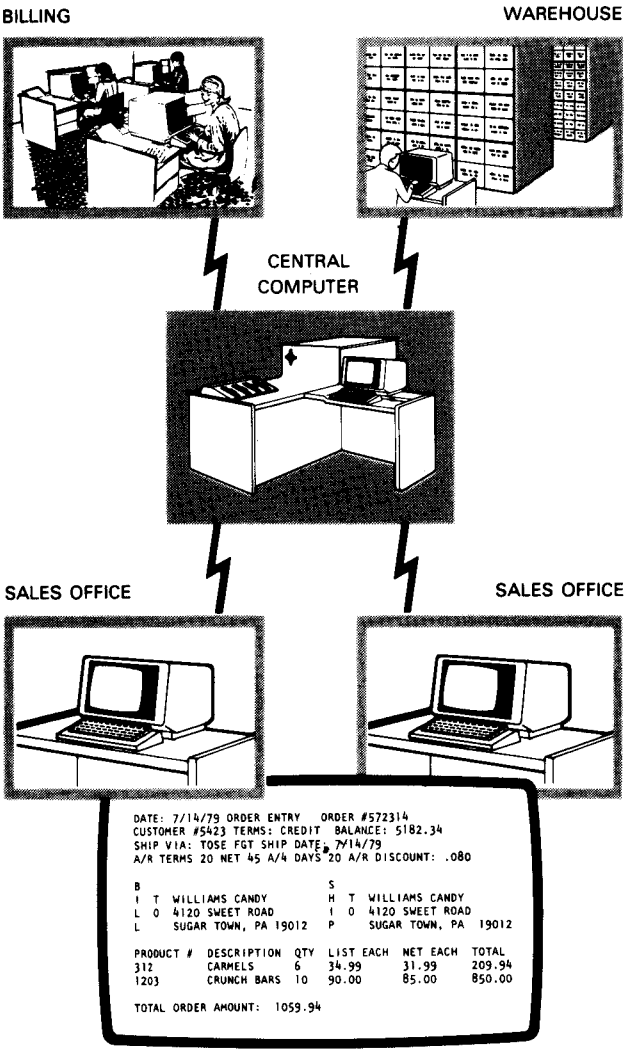
- Billing System

  Money tied up in accounts receivable doesn't earn money. The faster you do your billing, the faster you receive your profits. Transaction processing speeds billing.

- Sales Order Entry System

  The big advantage of a transaction processing sales order entry system is speed. Your salesman tells your customer within seconds if an item is available and when you can ship it. The customer knows what the order will cost, including discounts and shipping. The warehouse receives a shipping bill while billing gets a copy of the order. All of this is done within seconds, not the hours or days it would take without sales order entry.

Order Entry System



One of Sperry's most popular applications used with ICAM is a transaction processing system called the Information Management System (IMS). Like the example on the following page shows, it's conversational in nature; the terminal operator carries on a dialog with IMS, asking IMS for information. But that's just the beginning of what IMS gives you.

IMS is a system with more flexibility that most users ever need.

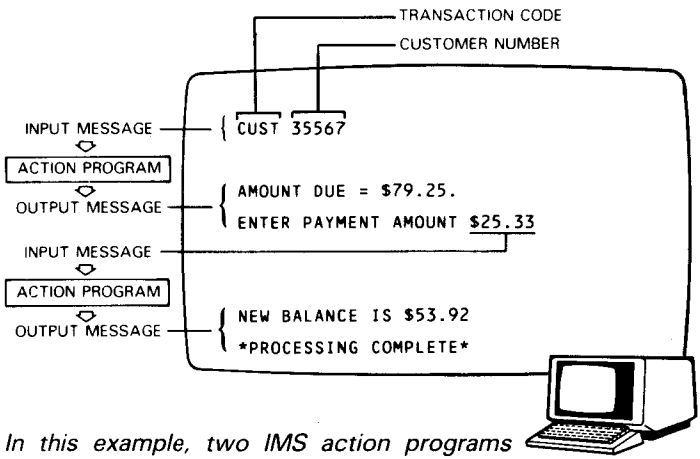
- Do you want only authorized people to see data or to be able to change data in a file?

- Do you need data formatted in a specific way or more than one application to be processed concurrently?...Say both sales order entry and inventory control!

IMS can do all of this and more.

To perform a file function with IMS, you key in a short message at your terminal or workstation that includes a transaction code. This code calls a specific action program. The action program performs just the functions you want and responds with an output message. Thus, you have quick and easy access to your files without complex programming or the wait required with batch processing.

IMS Operation

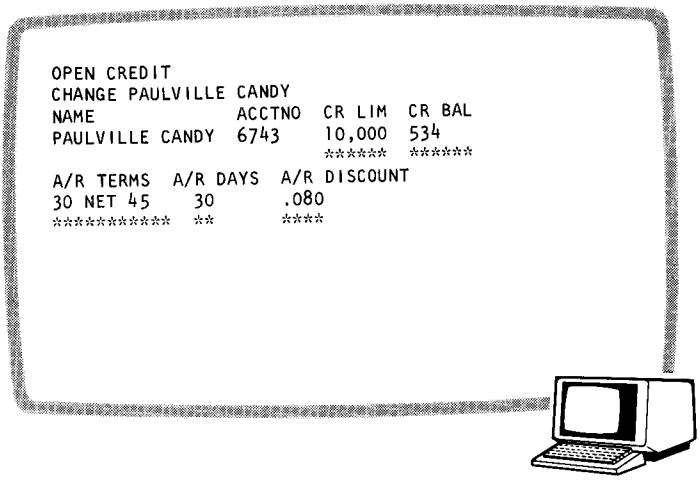


In this example, two IMS action programs are sequenced to produce AMOUNT DUE information, allow data entry, and compute a new balance for a specific customer account.

UNIQUE Language

IMS comes with a built-in, easy-to-use language called UNIQUE that is designed to let your terminal operators work with your data files. The following screen is an example of the display of how to change the contents of a file using UNIQUE.

Using UNIQUE



UNIQUE is a powerful language, but because it is designed to serve many different kinds of users, it is necessarily general in nature and might not fit your exact needs. Therefore, you might want to write your own action programs within IMS to fit your own specific needs.

IMS gives you a choice of three programming languages in which to do this:

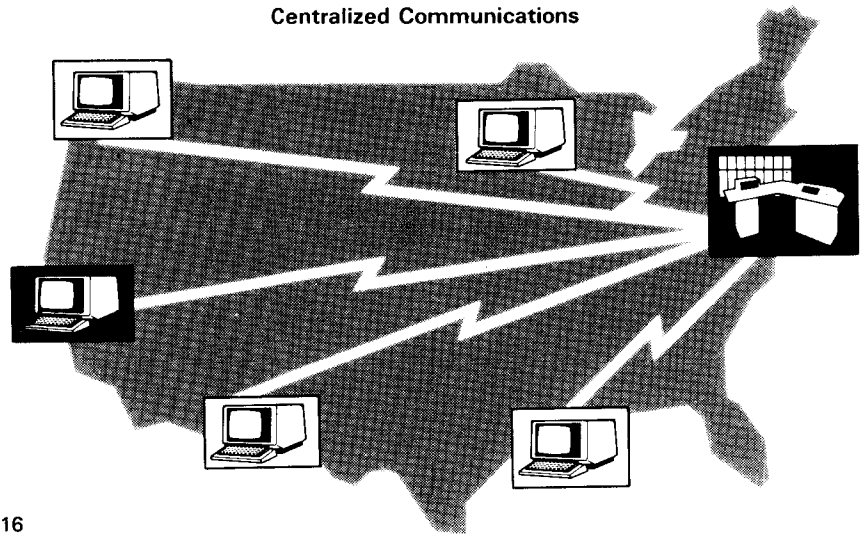
- COBOL
- RPG II
- BAL (basic assembly language).

distributed data processing – multiplying the power of your computers

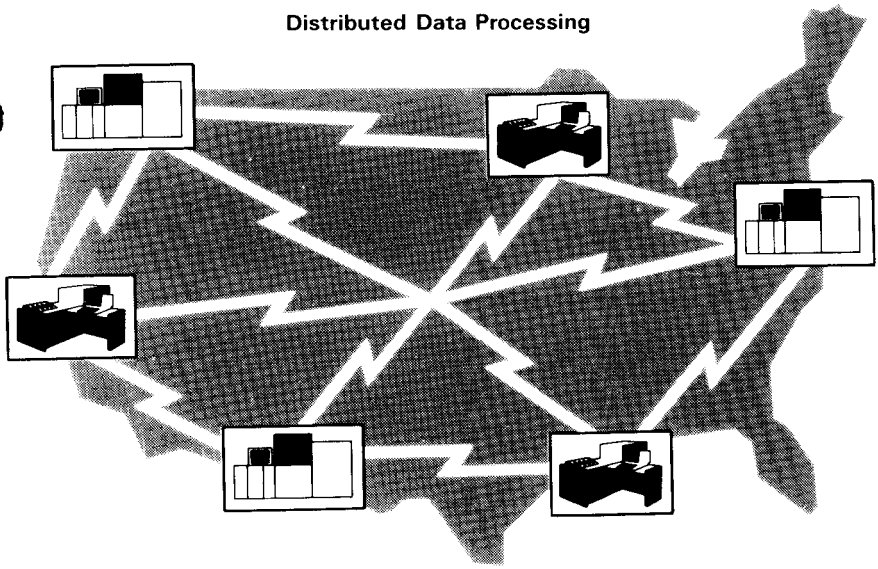
Let's assume your company has plants all around the country and they all require computer services. You could purchase a single large computer for all the plants to share. However, this has one obvious disadvantage – every location must use the same computer.

For example...Suppose you are a branch manager in San Francisco with your headquarters (and the central computer) in New York. To use the computer, you must send your raw data through your terminal to the computer at headquarters; then, reports (as processed data) must be sent back to your location.

Centralized Communications



Distributed Data Processing



To keep telecommunications costs down, you might need to communicate at night, and you may have to wait a day or more for reports.

What happens if the home office computer is down for a day!

The significant fact is that you don't control the most important management tool of all – your own data.

Distributed data processing (DDP) provides the answer. In a DDP system, independent computers are linked so that any computer can access another computer's data or send jobs for processing.

The DDP System

Just as administrative responsibilities in a large company are distributed over a number of managers, jobs and data are distributed over a number of computers in a DDP system.

Managers coordinate their work in your company to promote efficiency through communications. Similarly, computers in a DDP system promote efficiency by coordinating their work to process jobs and data on any computer in the DDP system.

Three DDP products are currently supported:

1. DDP Transfer Facility

   The transfer facility allows you to transfer files and jobs between computers. You can also inquire about the status of a job or a file on a remote computer.

2 DDP File Access Facility

The file access facility enables you to access and process files residing on remote systems and write application programs to initiate and communicate with them. The programs must reside on different computers.

3 IMS Transaction Facility

The information management system (IMS) transaction facility lets you process IMS transactions at a remote computer.

interactive programming – talking with your computer

Formerly, you wrote a program, keypunched it, and submitted it to the computer center. After a few hours – or days – you got back the results. Then you started the process over again – write the corrections, keypunch them, resubmit the job.

Sometimes, this is still the best way to go, but more and more interactive programming is replacing it. Now you "talk" directly to your computer from a terminal. Gone are the keypunches, the trips to the computer center, and the waiting.

SPERRY's OS/3 software operating system provides interactivity through a product called interactive services and, through ICAM, links your terminals to it. Interactive services offers many features, but the most important is the general editor. The general editor provides a comprehensive set of commands for creating and maintaining computer-based files on disk and diskette.

GENERAL EDITOR

```
ED000 EDITOR VERSION XX.X READY
1.0000▷DATA LINE
2.0000▷@DELETE 1
2.0000▷
```

The general editor also contains special language editor subroutines that simplify the creation and maintenance of RPG II and COBOL programs.

COBOL EDITOR SUBROUTINE

```
OS/3 EDT/COBOL              COBOL EDITOR(V8.0/1)-Selective Creation Mode
*************************************************************************
                  Standard COBOL Coding Form        Line nnnn.nnnn
 CA  B
  1.0000      IDENTIFICATION DIVISION.
  2.0000      PROGRAM-ID. LABELS.
  3.0000      ENVIRONMENT DIVISION.
  4.0000      CONFIGURATION SECTIONS.
  5.0000      SOURCE-COMPUTER.UNIVAC-OS3.
 Continuation Cde (NRM)  [Next Screen is the Standard COBOL Coding Form)
  NRM = Normal Continuation     TMP = Display Creation Screen List
  CMD = Enter EDT Command Mode   sss = Display Creation Screen sss
  RET = Return to Ordered Mode   vvvvvvvv = Display Verb Skeleton vvvvvvvv
EDT Command:@
*************************************************************************
```

Editor Subroutines

With these editor subroutines, your programmers sit down at a terminal, key in their programs, and write them directly into system files. If they make a mistake or want to make a change, they fix it or simply type in the new material. When they finish writing the program, they instruct the computer to run it and the results are returned to your terminal.

Total time for the program cycle? Writing time plus execution time. With interactive services and the editors, your programmers do more in less time, as you can see.

the network – distributing your terminals

The one part of a communications system that is unique to each user is the *network of terminals.* You can use the same kind of terminals that a thousand other businesses are using but how you arrange your terminals into a network is as unique as your business.

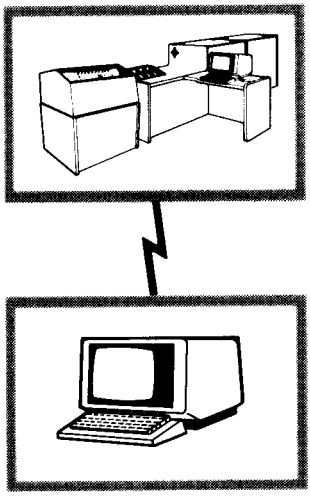
Perhaps you have a dozen sites that need remote batch processing. Maybe each of your 17 sales offices needs to be tied into a sales order entry system. Possibly you need 20 terminals in billing, 5 in customer relations, and 15 in accounting. No matter what network arrangement you need, you can have it with ICAM.

In any communications system, there are five basic ways to arrange terminals and data transmission lines. ICAM supports them all and allows you to mix and match them. They are:

- Point-to-point Network
- Multipoint Network
- Star Network
- Hierarchy Network
- Ring Network

The simplest network arrangement, point-to-point, has just one terminal on a line. With some terminals, point-to-point is the only way you can go. In other cases, point-to-point best meets your needs. Maybe you have a number of sales offices that need just one terminal each.

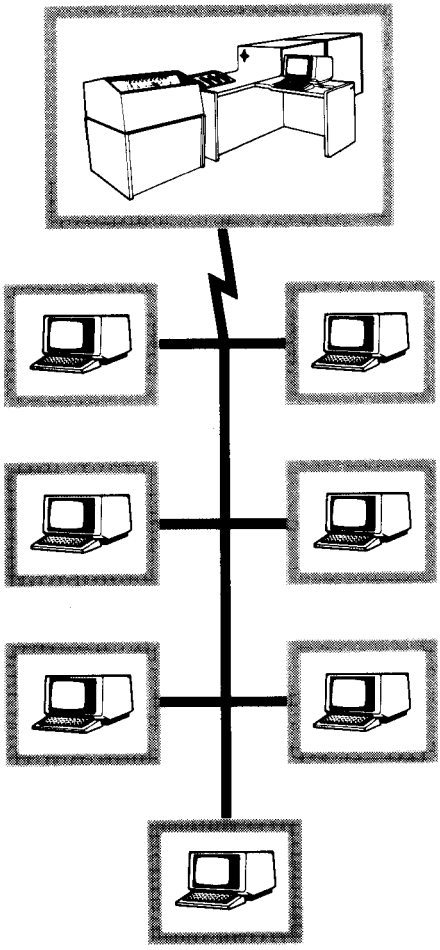
Point-to-point Network



When you have more than one terminal in a location, point-to-point is expensive. A data transmission line can carry information thousands of times faster than your operators can type it in

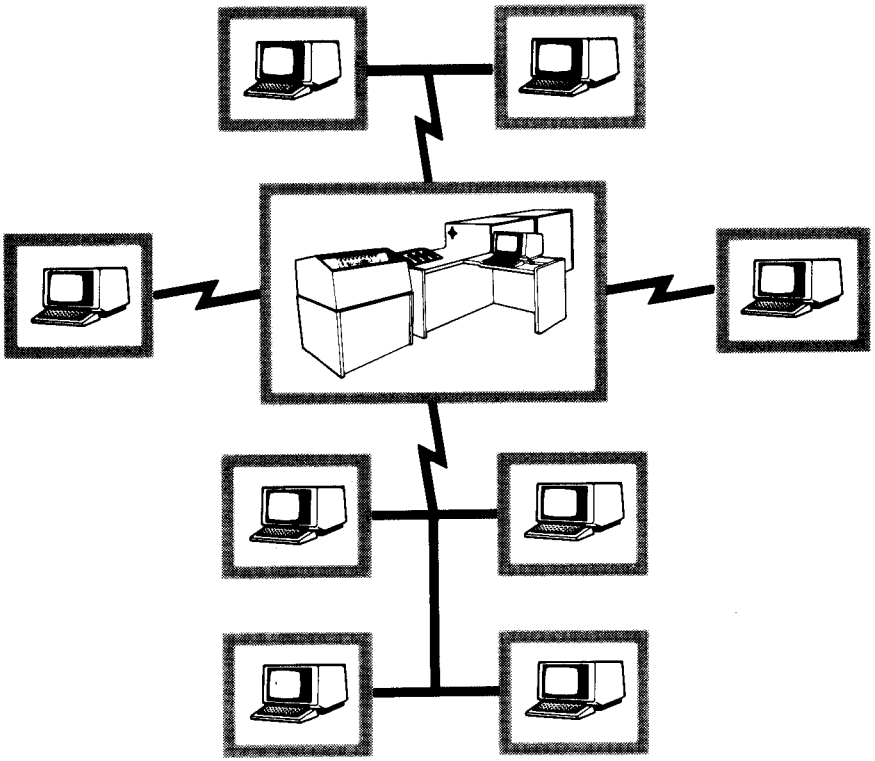
or read it. By giving each terminal its own line, you're paying for many times the communications capacity you need. With multipoint lines, several terminals share a single line, giving you better use of the resources you're paying for.

Multipoint Network



A star network combines two or more point-to-point or multipoint lines into a network.

Star Network

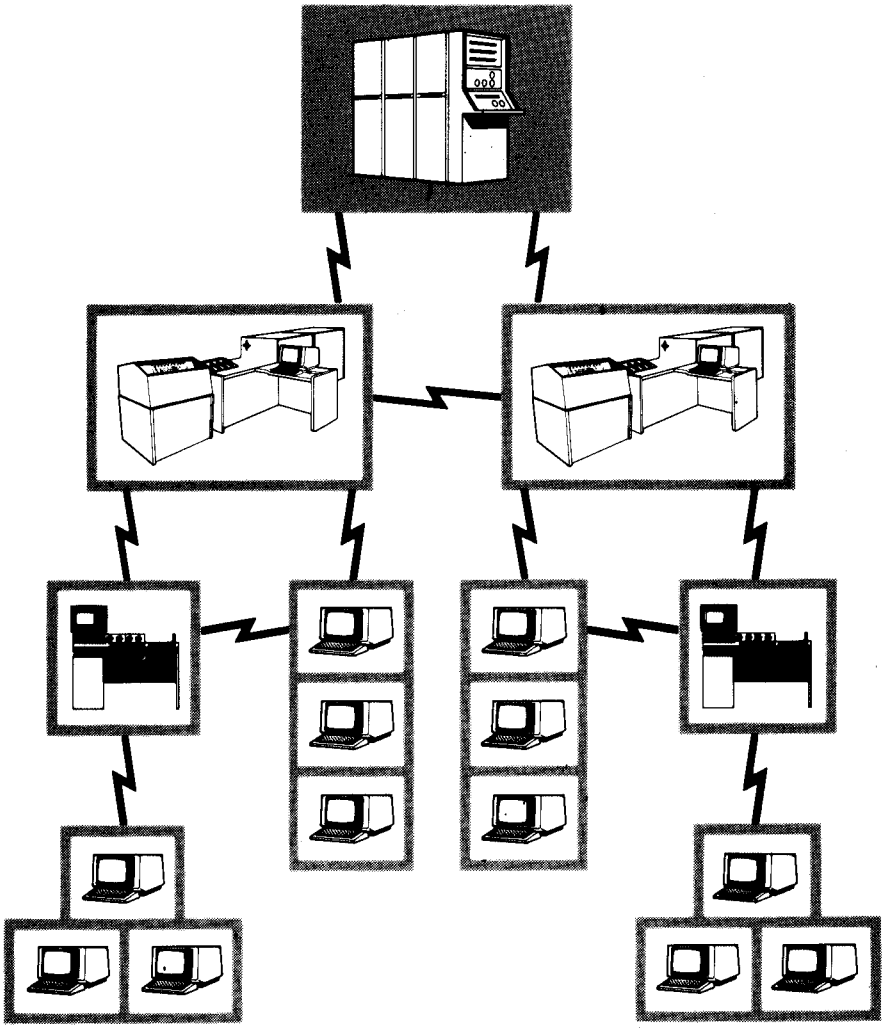


In the three networks discussed so far (point-to-point, multipoint, and star), a central computer controls the system. In the two other types of networks, several computers share the control. One arranges the computers in a *hierarchy.*

Each computer in a hierarchy communicates only with its immediate supervisor and its subordinates; the terminals tie into the lowest levels of computers.

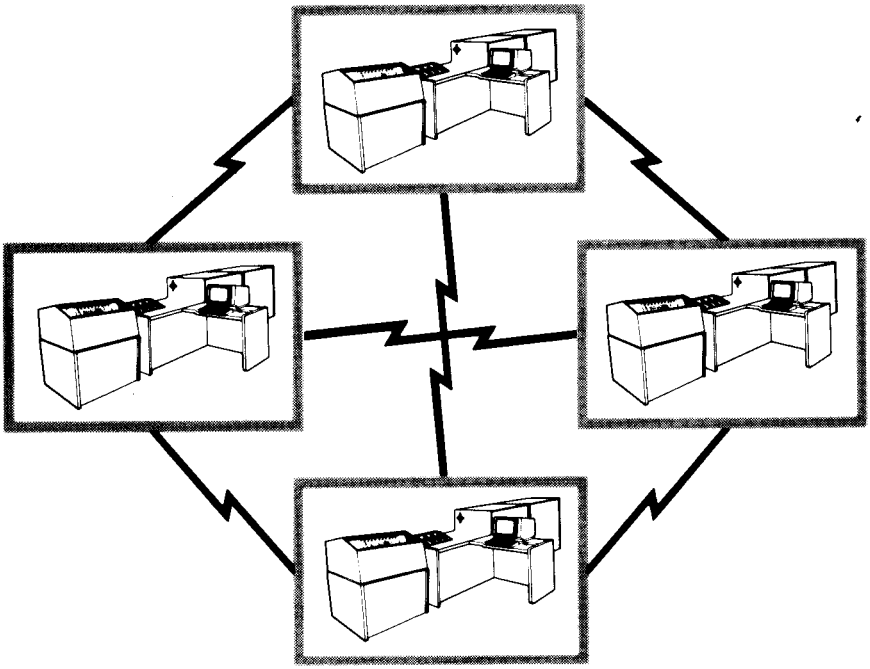
Suppose your company has a main office, two regional offices, and sales offices within each region. The main computer is in the home office. Linked to it by data lines are two computers for the regional offices. Tied into the regional computers, perhaps through intelligent terminals or minicomputers, are the terminals in the sales offices.

Hierarchy Network



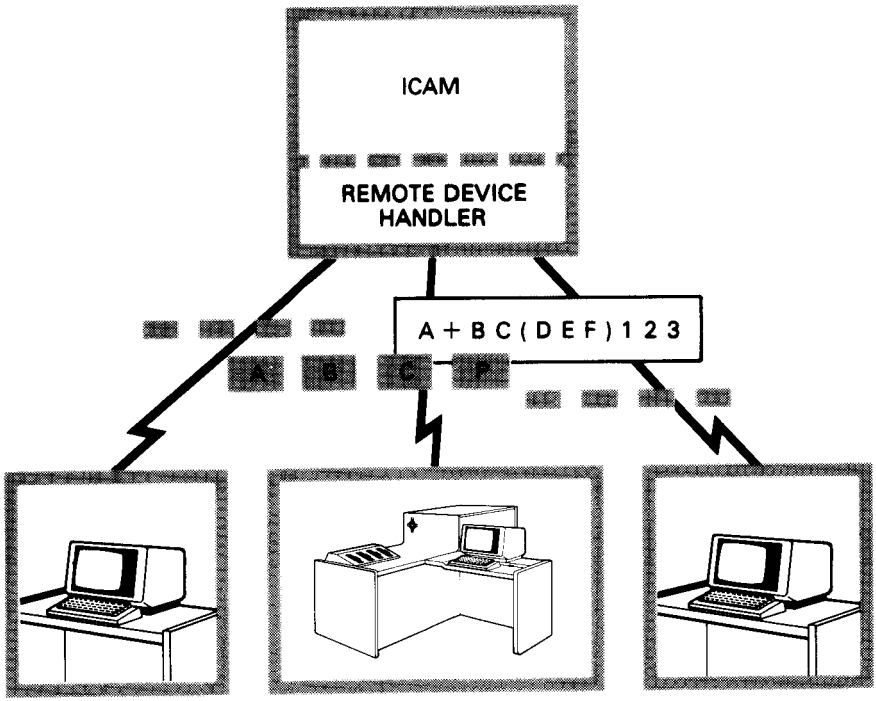
The second way to distribute control makes all the computers in a network equal by tying them into a ring. Even more freedom can be given the computers by linking each one to every other computer in the network.

Ring Network



the terminals – the hardware for communications

ICAM's support for your terminals is as broad as its support for your applications programs. With it, your computer can communicate with almost any kind of terminal or remote computer.



With this kind of flexibility, you can arrange your communications hardware into any kind of network you need. ICAM does this by providing code translation and remote device handlers to suit your needs.

Most computers process data internally in a code that is different from the one they use to send and receive messages over a communications line. This requires translation every time a message is sent to a terminal or a message is received from a terminal.

For example, a common standardized communications line code is ASCII*, and ICAM provides automatic translation to ASCII code whenever required. Often, however, terminals require slight variations to work correctly.

* *American National Standard Code for Information Interchange. A 7-bit code established by the American National Standards Institute, an organization for establishing voluntary industry standards.*

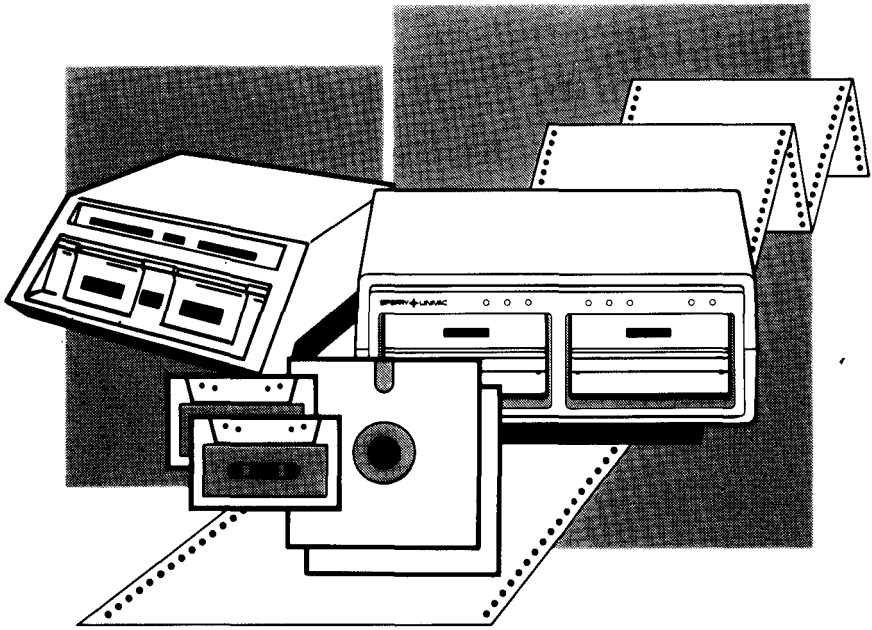
Translation Table

ICAM solves the translation problem by letting you create your own translation tables (or you can easily modify Sperry's) and include them into the ICAM software. You can use the same table for all the terminals on the same line or you can use a different table for each terminal. You can even have a different table for incoming messages and and outgoing messages.

Line Protocol

Line protocol is a set of procedures that must be followed on a communications line for data to be successfully exchanged between a computer and a terminal. These protocols are necessary to ensure that messages are accurately received and no data is lost. You can imagine how important line protocol is while printing paychecks!

There are as many different protocols as there are terminal and computer manufacturers. ICAM solves the protocol problem for you by providing a remote device handler for each type of communications device that Sperry supports. This covers a wide range of terminal types. If we don't cover your type, you can write your own and include it in ICAM.



Additional Flexibility

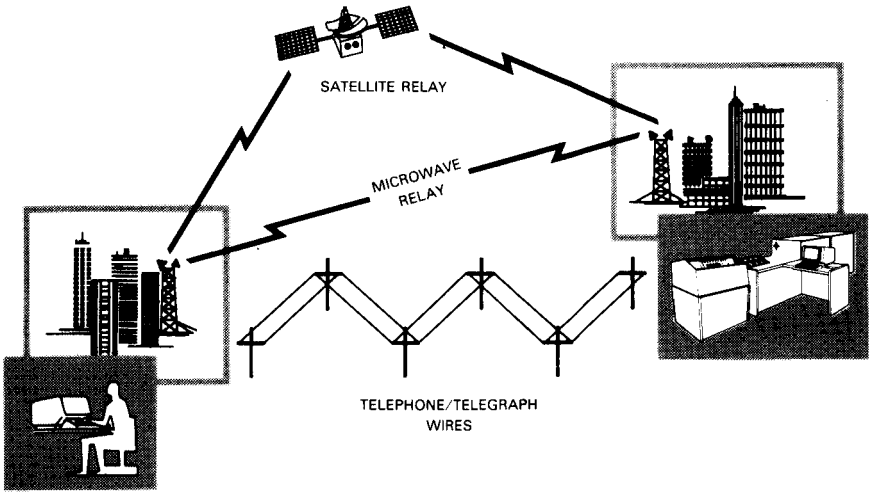
With many terminals, the terminal itself is just the beginning. You can add a tape cassette system or a diskette system to store large amounts of data. We also have printers that can be used with video screen terminals. Some terminals have optional paper tape units for sending or receiving batches of information.

the data carriers – tying the terminals to your computer

Data carriers – these are the communications organizations that provide the facilities to connect your computer and your remote terminals. The telephone and telegraph systems are the best known.

In recent years, data carriers have added new equipment designed to carry just computer data. These are systems such as the public data networks used in many countries. (See following discussion.)

The important point is that ICAM is flexible and you can choose the kind of support available from your local data carrier whether it is a national postal system or a private carrier.


SATELLITE RELAY
MICROWAVE RELAY
TELEPHONE/TELEGRAPH WIRES

public data networks – special carriers

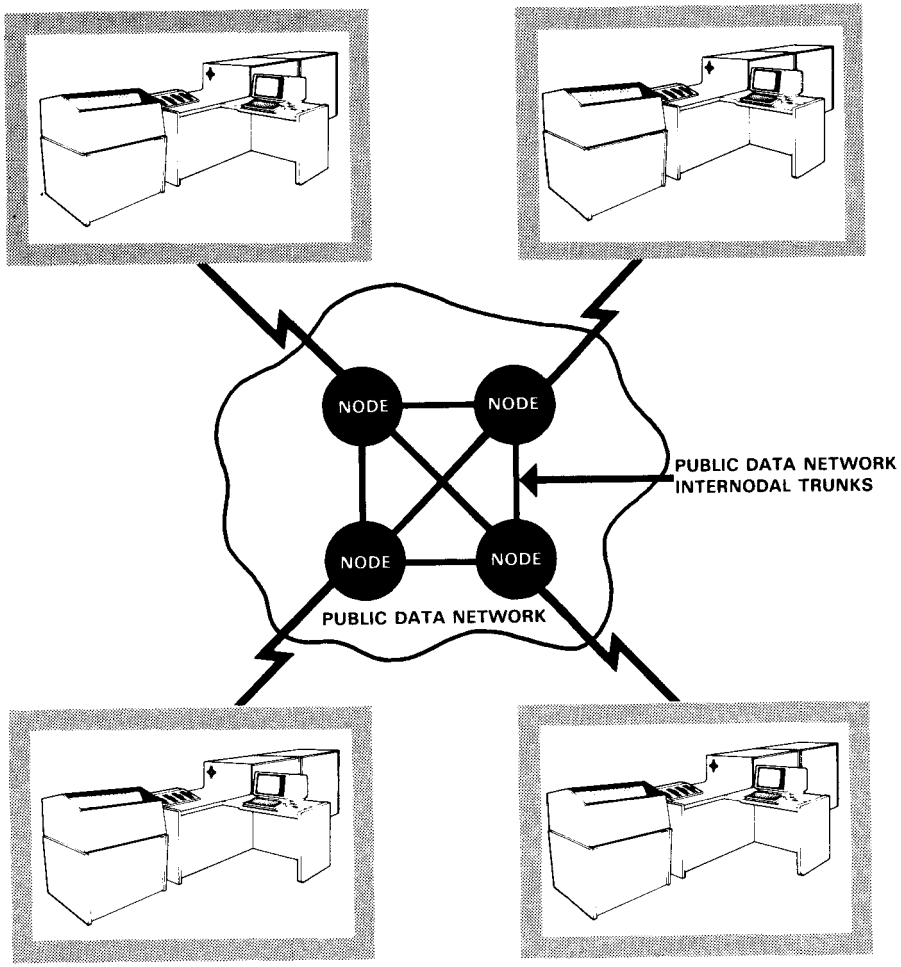
Most often the design of a communications network is governed by the geographical locations of the remote sites. Providing dedicated communications lines (those you rent for your own exclusive use) to outlying sites is expensive and dialed lines (switched lines you share with other users) may not be adequate to satisfy the need.

To solve this problem, a new type of communications facility has emerged throughout the world. It is called the public data network (PDN). PDNs are usually controlled by the postal telegraph and telephone authorities in each country.

PDNs provide facilities especially designed to handle high-speed digital communications. They achieve significant economies for their users by allowing them to share the high-speed trunks that interconnect the PDN communications centers.

The following illustration shows several System 80 computers connected by a PDN. Each System 80 computer connects to a PDN center (node) which in itself is a specialized computer.

Public Data Network (PDN) System



Two different types of PDNs exist – packet-switched and circuit-switched.

- A packet-switched PDN node divides each message you send into fixed-length units called packets, adds a header to identify each packet and its destination, and routes them over the optimum trunk. At the destination, the packets are reassembled and the message given to ICAM.

  ICAM makes the message available to your program on request, the same way that it provides any other message. Packet-switched PDNs usually charge on the basis of how many packets they deliver.

- Circuit-switched PDNs provide temporary private-line-like connections between your computer and a remote site. The main difference between circuit-switched service and a private line is that a connection is quickly made only when it is needed. The line is immediately disconnected when it is no longer needed. You pay primarily for the time you use the circuit.

ICAM supports most popular packet-switched and circuit-switched PDNs world wide. Configuring an ICAM network to support communications to a public data network is just as easy as configuring an ICAM network to support an ordinary telephone line.

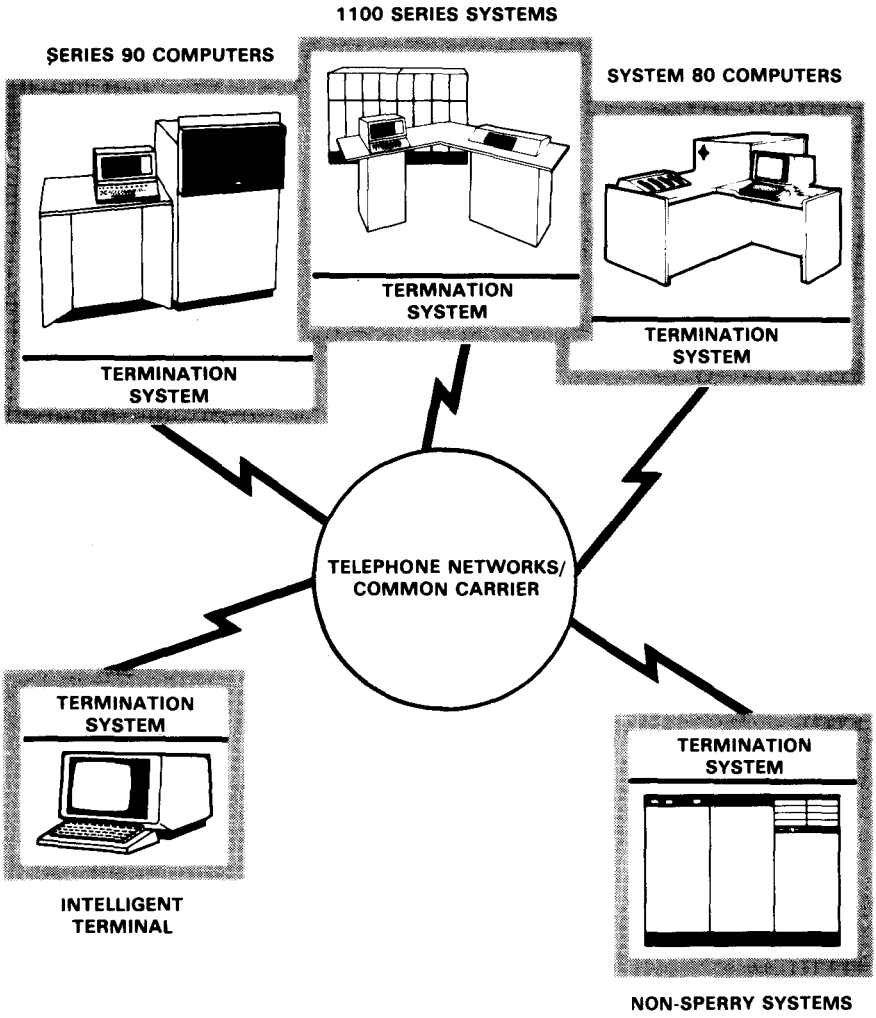
distributed communications architecture – modern design for communications

Distributed communications architecture (DCA) is Sperry's way of providing you with the advantages of new communications techniques and facilities.

With DCA, you can communicate with many different remote users without knowing where they are physically located or even the kind of equipment they have. DCA protects the integrity of your data, chooses the best route for the data to take, and controls its flow so that communications lines are not overloaded or under-utilized.

ICAM provides DCA support for the Sperry OS/3 operating software system through a program called the DCA termination system. The termination system speaks a common language used by all members of the communications system. Your programs "talk" to ICAM just as always. DCA supports host processors, intelligent terminals, and non-Sperry systems (provided a termination system is written for them).

An important product supported by ICAM that uses DCA is distributed data processing (DDP). DDP is described separately in this booklet.



DCA Communications System

ICAM – what it is and how to create it

Now that we have told you all the good things that ICAM can do for you, let us say a few words about ICAM itself and how you go about creating it.

ICAM is a special kind of program called a symbiont. It works with the Sperry OS/3 software system supervisor (the program that controls the overall operation of OS/3). The sole purpose of ICAM is to support communications. You load ICAM when you need communications; you terminate ICAM when you no longer need communications.

Once you've tailored ICAM to your needs, it is almost invisible. Your programs pass messages to it and receive messages from it; but that's no more than they do to send data to or get data from a disk file.

We've talked about tailoring ICAM to fit your system, and that might seem just a bit complicated. Well...in most cases it really isn't.

About 90 percent of an ICAM definition is a simple listing of the communications devices and services you want to use. The rest is telling ICAM how many system resources it is allowed to use (usually only two or three program statements).

You create ICAM by filling out a set of statements (writing program coding) and submitting the statements to the OS/3 system generation process. System generation processes your statements and automatically selects the code needed. It even checks your statements for errors. If you make an error, it *flags* the error so you can fix it before you proceed any further.

The sample coding on the following page (a computer program) is all you need to create a communications system to service a single communications line that serves two terminals.

Sample Coding:

```
      1         10        16                                    72
1.    NET1      CCA       TYPE=(STDMCP)
2.              BUFFERS   50,256,10,ARP=9
3.    LNE1      LINE      DEVICE=(UNISCOPE),                    X
                          CALL=5423539,                         X
                          TYPE=(2000,SWCH,SYNC),                X
                          INPUT=YES,                            X
                          HIGH=MAIN
4.    TRM1      TERM      ADDR=(28,51)                          X
                          FEATURES=(U400,960)
5.    TRM2      TERM      ADDR=(28,52),                         X
                          FEATURES=(U400,960)
6.              ENDCCA
```

Explanation:

1. Names this network and specifies the type of interface your program will use to communicate with ICAM.

2. Gives requirements of buffers to temporarily hold messages.

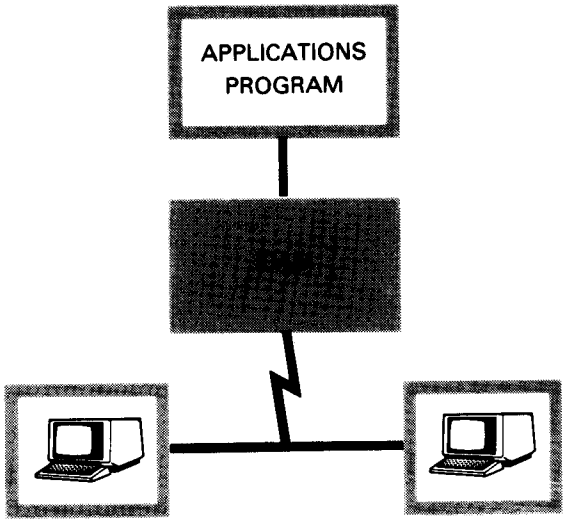
3. Gives type of terminals on the line, telephone number of line, characteristics of line, and defines input and output queues.

4. Defines characteristics of first terminal.

5. Defines characteristics of second terminal.

6. Ends ICAM definition.

The coding on the opposite page defines an ICAM to support the network shown below.



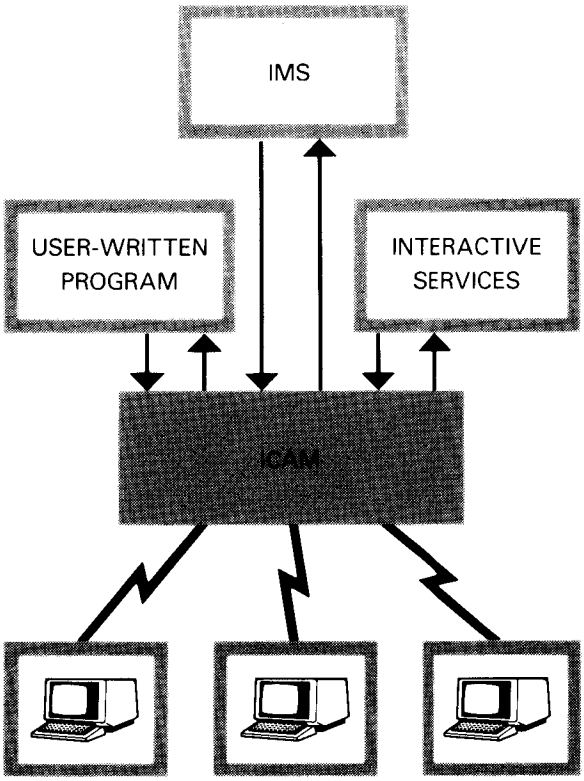
Just as the post office has different services for different customers, so does ICAM, except we call them interfaces. Equivalent to first class delivery, we have the *standard message control program interface.* It gives you the most support, making your programming easier.

Almost identical to it, the *transaction control interface* supports the Information Management System (IMS).

ICAM's equivalent of bulk rate is the *direct data interface.* It's used for transmitting large amounts of data for remote batch processing and distributed data processing systems.

After you create ICAM, you need an applications program because "mail is not usefull useless there is someone to read it!" The applications program provides this *reading service.*

You can use a Sperry-provided applications program such as IMS or interactive services, or you can write your own. If you decide to write your own programs, ICAM allows you to do it in any of three common computer languages. The first two, COBOL and RPG II, are designed for business applications. The third, basic assembly language (BAL), is the working language of the computer.


IMS
USER-WRITTEN PROGRAM
INTERACTIVE SERVICES
ICAM

ICAM – summary

ICAM is flexible. With it, you create a communications system that extends the reach of your computer to every part of your business. You decide what types of applications programs you need to solve your problems – ICAM will support them. In many cases, you can use one of Sperry's applications packages. Or your programmers can write the applications programs for you in any of three computer languages with no more trouble than if they were preparing any other program.

Once your programs are ready, you put together a network to use them – ICAM will support it. ICAM can work with almost any terminal available whether it's supplied by Sperry or another vendor. And ICAM allows you to arrange your terminals into any network configuration you need.

Because ICAM is flexible and modular in design, you get a tailored program that exactly meets your needs – no more, no less.

But most of all, ICAM lets *you* be flexible and meet the demands of a changing world.