11

MAINFRAME CONNECTION

INTRODUCTION

A major benefit of networking is that microcomputers and the work performed on them is no longer are isolated. Information can be passed from one computer to another and read and updated as needed. The process of linking PCs and combining their capabilities creates a flexible new computing resource. In the corporate environment the next step is to link this new resource with an established one: the mainframe computer.

It is likely that at some point you will need to communicate with one of the larger computers, called mainframes. Perhaps it will be a small departmental computer used by your work-group. It may be it be a larger one, run by your division's MIS department. Or it might be a corporate computer thousands of miles away. Herewe will discuss:

* reasons for communicating with a mainframe.
* a brief history of IBM's data communications.
* methods of connecting personal computer networks to larger computers.
* problems that can occur.
* prospects for the future.

The IBM PC first entered the corporate world by the back door. The PC was quietly brought in on purchasing orders for typewriters. And when it arrived, most of the work the PC was given probably could have been done with typewriters, calculators and pencils and paper.

Quickly, the PC began to achieve respectability. First-rate application programmes came along for word processing and spreadsheets. These programmes were easy to learn and use. Often they were a distinct improvement over comparable mainframe programmes. But the PC remained isolated from the mainframe and unable to access corporations' primary data banks.

Then devices began to appear that let the PC communicate with the mainframe and manipulate mainframe data (see Figure 11.1). PC-to-mainframe communication was a prerequisite to making the PC part of large-system computing. Now that the PC is established, the door is open to explore fully the other applications of an intelligent desktop workstation in a mainframe world.

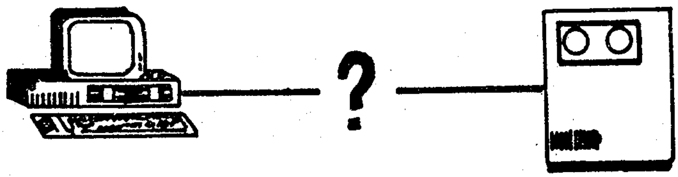

Fig. 11.1 : Micro to mainframe

Incidentally, PCs can communicate with larger computers made by manufacturers other than IBM. Some of these non-IBM computers function so much like IBM's computers that they are known as plug-compatible computers, which means you can plug anything into them that you can plug into an IBM mainframe. Because plug-compatible computers are essentially equivalent to IBM's, anything written here about IBM mainframes applies to plug-compatible computers as well.

WHAT IS A MAINFRAME?

For the many computers that are not plug-compatible with IBM, there is a thriving business in hardware and software that lets IBM computers and non-IBM computers communicate. This will doubtless continue to be true for local area networks. So if your mainframe is not IBM or IBM-compatible, you will need to check with various dealers and users for details on the products available and how they interface to the IBM network. However, much of the information here is still applicable even for mainframes that are not IBM-compatible.

Not long ago, there were three distinct types of computers: mainframes, minicomputers and microcomputers. Microcomputers could support one person, mainframes could support hundreds of people, and minicomputers fell somewhere in between. IBM so dominated the mainframe market that to some people "mainframe" meant "made by IBM". Until recently, however, IBM had few products in the minicomputer category.

Now, demarcation is more difficult. Computer power has increased dramatically. Some children's toys have more computing power than ENIAC, the first computer, which filled an entire room and dimmed city lights when it was switched on.

Similarly, in many cases today's so-called "personal" computers are more powerful than yesterday's minicomputers and mainframe computers. Some personal computers are now so powerful that more than one person can use them at a time. So simply defining a mainframe as "any computer that can support more than one user at a time" is no longer sufficient.

For the purpose of discussion, we will define a mainframe as a multiuser computer running an IBM proprietary operating system. Computers running MS-DOS or a form of UNIX are not included in our definition of mainframe because IBM does not own the rights to these operating systems. After all, if the mainframe ran MS-DOS, it could be connected through the LAN just like a big PC. In a similar way, a UNIX computer usually cannot communicate with MS-DOS unless a special hardware bridge product is used.

Some might object that this definition of mainframe includes smaller computers such as IBM Series/1 and IBM System/36 mainframes, which are often referred to as minicomputers. Rather than get bogged down in a discussion of the dividing line between minicomputers and mainframes,we will avoid the problem by calling all of them mainframes, unless we are referring to a specific computer model. Here, we are more concerned with which computers can communicate with each other, rather than with their relative capabilities.

WHY TALK TO A MAINFRAME?

When the notorious 19th-century criminal Willie Sutton was arrested, legend has it that someone asked him why he robbed banks. He replied, quite logically, "Because that's where the money is". Similarly, people need to communicate with mainframes because, in most cases, that is where the data is. Despite the technological advances of personal computers, mainframes are unsurpassed in their ability to manipulate huge amounts of data. You can think of a mainframe as a giant disk server, providing a large amount of hard disk storage that can be shared.

Whereas an IBM PC/AT/486, for example, can support a single 300- megabyte hard disk drive, a mainframe can easily support over six 1000-megabyte drives. And, as many users are discovering, a single file, such as a large database, can easily fill an entire 300-megabyte drive. No matter how powerful a PC is, it simply cannot support that amount of data. Moreover, mainframes are most often used by large corporations, so many people will want access to the mainframe's vast amount of data storage. Although no PC can support more than a few users, a large mainframe can support dozens, even hundreds, of users.

In the same way that a mainframe can act as a large disk server, it can also act as a large print server. But instead of supporting a $5,000 laser printer, as a PC print server might, a mainframe would support extremely expensive peripherals. For instance, IBM makes a large, high capacity laser printer that costs over $100,000. Other peripherals might be typesetting machines, flat-bed plotters several feet across, or line printers that print hundreds of lines per minute.

Mainframes also have much more computing power than PCs. Computing power is measured in MIPS, or million instructions per second. A PC can usually perform around one MIPS, but a mainframe can perform ten or more MIPS. The difference in power is not significant if you are balancing your checkbook, but it can make a big difference if you are calculating a complicated equation.

Mainframes can also make life easier for PC users by automatically backing up files. Nearly everybody has forgotten to make a back-up copy of a file. If you are lucky, you only lose a few minutes' work: if you are not, you lose 300 megabytes of data that took weeks to accumulate. With a connection to a larger computer, you can set up a system whereby PC files are automatically backed up onto the mainframe.

Finally, a mainframe can act as a gateway to another mainframe or network. It is analogous to large airlines. You do not fly from Calcutta to Srinagar directly; instead, you might fly from Calcutta to New Delhi, then from New Delhi to Srinagar. If your destination is less frequented, like Leh/Ladakh, you would break up the flight even further by flying from New Delhi to Srinagar, then from Srinagar to Leh/Ladakh. It is the same with gateways. You do not have to communicate directly with a particular computer; it is sufficient to communicate with another computer that can, as shown in Figure 11.2.

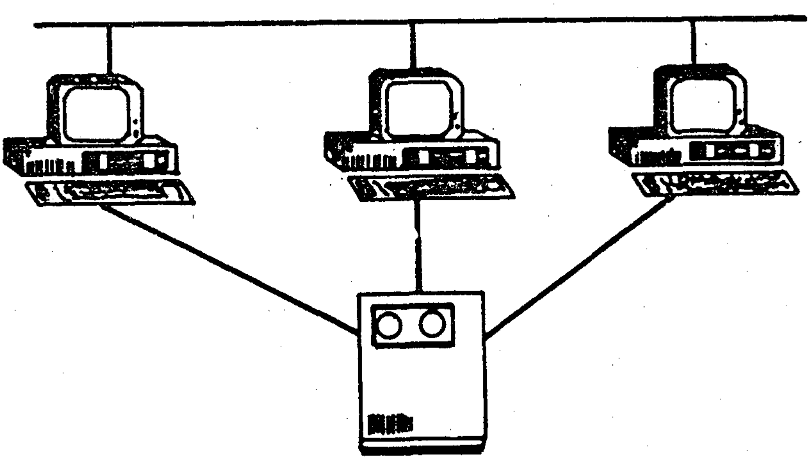

Fig. 11.2 : Terminal emulation

BRIEF HISTORY OF IBM DATA COMMUNICATIONS

To understand the techniques used for communicating between PCs and mainframes, we need to see how IBM data communication developed. In the early 1970s, when data communication began to develop, there were two kinds of devices: computers and terminals. Terminals functioned only as input/output devices; they could not store information or manipulate it in any way. Consequently, all data communication were set up in a hierarchical fashion. The computer was the "master" and the terminal was the "slave". The master computer started and stopped all transmission, and governed what information was sent. The slave terminal could respond only if information was coming too fast, or if the transmission was garbled.

The master-slave hierarchy was fine back when terminals truly had no computing power (sometimes referred to as intelligence; terminals with no computing power are called dumb). Few computers had to communicate with intelligent devices such as other computers. When they did, one computer would revert to a slave status (usually through a programme) to accept the data.

But as the industry matured, computers in a range of sizes became available. Small terminals became as powerful as the master computers. Large organisations, and organisations with many branch offices, found they wanted to transfer information between their computers more often. The master-slave relationship was limiting, and wasted the power of the slave computer.

The proliferation of devices spawned another problem. IBM's data communication products like VTAM and SDLC had been gathered under the umbrella term of SNA (Systems Network Architecture). Much of SNA consisted of programmes developed specifically to communicate between two particular devices. Each combination of new devices required a new set of rules (a protocol) to specify how the two devices would communicate.

Consequently, every time a new device became available, it had to imitate an existing device or IBM had to write a series of data communication programmes following a new protocol so that the new device could communicate with existing devices.

Then came the introduction of the personal computer. Here was a device with significant computing power, but so small and relatively cheap that it blossomed on desks everywhere. People wanted to send data from their PCs to the mainframe, and take data from the mainframe for further manipulation on the PC. But mainframes could only communicate with a PC if the PC "pretended" to be a dumb terminal (terminal emulation). So users had a choice. They could print the data from the mainframe and then retype it on the PC. Or they could write a programme to transfer data between applications. Unfortunately, each combination of applications required writing a different programme.

LU 6.2 - The Final Solution

As might be expected, users were not thrilled about either of these options. They wanted to transmit data directly from the mainframe to their PC applications. IBM's solution was to develop one last set of protocols. Unlike other protocols, which were device specific, this protocol set defined a way that two intelligent devices could communicate; neither had to be a slave to the other. This kind of communication is called peer-to-peer communication, because the two devices communicate as equals. Either machine can start, stop and control transmissions.

The protocol set is marketed by IBM under the term APPC, for Advanced Programme-to-Programme Communication. Following traditional IBM practice, APPC consists of one physical unit (PU 2.1) protocol and one logical unit (LU 6.2) protocol. These terms equate roughly to hardware and software; thus, the PU 2.1 protocol essentially defines peer-to-peer communication between actual hardware devices, and the LU 6.2 protocol defines peer-to-peer communication between the software and application that run on the devices. Users whose computers are connected to a network running APPC can write programmes that can communicate on a peer-to-peer basis with each other.

Specifically, LU 6.2 defines a group of verbs (subroutines) that programmers can use for communication. Programmes using these verbs have a common interface. The programe can perform any function, run on any computer and be written in any language, as long as it uses the verbs in the accepted way. The advantage of using common verbs is that each programmer does not have to programme the actual connection; IBM's communication products take care of the actual connection.

Just defining these protocols does not solve the communication problem. Because programmes do not use these protocols as yet, users cannot usually communicate directly with mainframes. The act that they have been defined, however, means that peer-to-peer software can be developed and will be available. In the meantime, terminal emulation is still the most common method of communication between PCs and mainframes.

OFFICE SOFTWARE PROTOCOLS

Another set of protocols is used between IBM's office software products rather than between hardware devices. DCA, Document Content Architecture, specifies the internal structure of a document, or how the document is stored in the computer. DIA, Document Interchange Architecture, specifies how documents can be moved, stored and retrieved.

The product DISOSS (Distributed Office Support System), follows the DIA and DCA standards. It uses a central mainframe computer to store and distribute documents. With DISOSS, users can communicate with users on other computers through electronic mail, store documents in a "host library", and retrieve documents for later use. Eventually, the distribution portion will be replaced by SNADS (SNA Distributed Services). The advantage of SNADS is that it does not require a central computer; instead, as with APPC, users can send, file and retrieve documents on a peer-to-peer basis.

TERMINAL EMULATION

Host computer systems, whether they are mainframes or minicomputers, are designed to support workstations commonly called "dumb" terminals. A dumb terminal does none of its own application processing, but must rely on the host system for its intelligence.

The simplest way for a PC to access a mainframe system is by emulating a dumb terminal. In emulation, one device assumes the characteristics of another device. That is, after an emulation package is installed in a PC, the PC becomes a functional replacement for the terminal. Emulation may require software only or a combination of hardware and software.

Several types of terminals, including the DEC VT100 series and the IBM 3101, are widely used. Suppose that a host system is running an application that interfaces with VT100 terminals, in this case, mainframe access can be easy. Communication software containing a VT100 emulation programme is loaded at the PC. A connection is made to the host through dial-up telephone lines, and the PC, which now emulates a VT100, can access the host as a dumb terminal.

One of the most popular types of mainframe terminal systems is the 3270 series from IBM. The 3270 terminals are used in on-line (interactive) sessions with an IBM host computer. These highly functional terminals support such features as text insertion and deletion and automatic cursor movement.

Through emulation hardware and software, the PC workstation can perform the same functions as the 3270 terminal. In addition, the PC can receive and store data from the host computer, modify or reformat display data, run local application programmes, and send the output to the host.

The host can be an IBM System/370, IBM 308X or 43XX processor. The PC can be connected via coaxial cable to a channel-attached IBM 3274 or 3276 cluster controller, or remotely to a bisynchronous or SNA/SDLC 3274 cluster controller (see Figure 11.3).

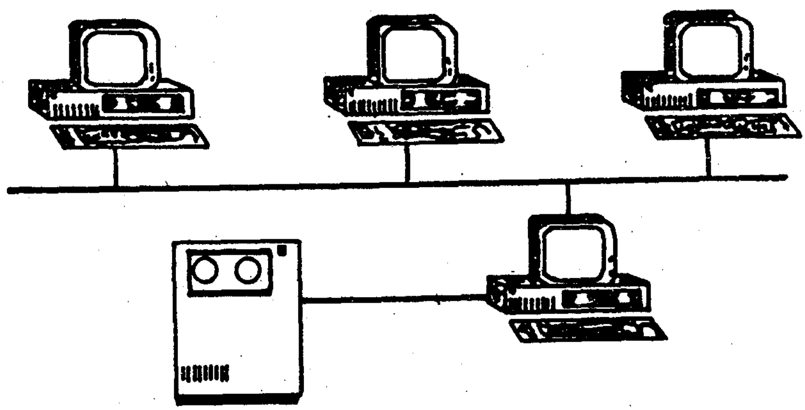

Fig. 11.3 : Gateway connection

A 3270 emulator is a type of gateway. Through it, virtual circuits are established between host and terminal devices, and go to a bisynchronous or SNA network and the IBM host. The emulation package includes a circuit board, which supports the physical connection between the PC and mainframe system. Also included is software, which runs on the PC and provides its 3270 functionality.

Emulators are available to make the PC look like either the 3278 or 3279 terminal, although accommodations must be made for some models. The 3278 Model 2 display is PC compatible, with 24 lines by 80 characters. Model 3 has a 33-line display; Model 4, a 43-line display. You cannot put more than the standard 24 lines on the PC screen, so to handle large displays, the emulators enable you to use the PC's PgUp and PgDn keys to scroll the screen.

The 3279s are colour terminals. Again, emulation is not perfect because the 3279 resolution is higher than that of the PC colour monitor. The alternatives are either to use only the text mode of the 3279 and forego the graphics, or to purchase a graphics monitor.

The IBM PC keyboard provides all the special functions of the 3270 series keyboard, although some adaptation is needed because the numbers and positions of the keys differ. Many companies, including IBM, provide 3270 emulation packages.

HOW TO EVALUATE AN EMULATOR?

The first step in evaluating a 3270 emulator is to check that it is both hardware and feature-compatible with your system, beginning with the PC. All emulators will not work with all IBM PC compatibles. The emulator board should fit any slot and be used like any other circuit board. Other parts of your existing system, such as cluster controllers, may be "IBM-compatible" but not compatible enough to support a particular emulator without some modification.

After you check compatibility, the next step is to look at emulation quality and how it is achieved. All emulator designers make accommodation for the screen and the keyboard, but these designers tend to follow their own rules. Which designer's scheme is best is a matter of personal preference. As you evaluate the emulator, test each key of the emulator-equipped PC. You may notice considerable variation in what a key actually does, what the documentation says it will do, and what the comparable key on a 3270 does.

A third step is to evaluate the emulator software. This software must perform a number of functions, the first of which is to permit you to enter the emulation mode. From there, many of the emulators let you run concurrent PC and 3270 modes, controlled through a toggle switch. One key, designed as a toggle, shifts the PC environment back and forth between PC and 3270 modes.

The emulator software has an important feature, the transfer utility set, which appears with varying degrees of success on early emulators. An emulator should be able to bring files down from the host to the PC and return them to the host when processing is completed. The transfer utilities should support whatever environment is necessary. Thekey-point is you must make certain that the software supports not only the two-way transfer, but also the desired mainframe environment - TSO or VM/CMS or both.

How the information is displayed should also be taken into account. Compare the emulated display with the 3270 to see that status lines or comments, are the same and that all the characters are compatible. Some differences can occur in the character sets, so designers must choose representative characters.

The emulator should be totally compatible with your present system. You should be able to disconnect the coax and connector from the 3270 terminal and plug the cable directly into the emulator board on the PC. Then, when you turn on the system and start the emulator software, you will be connected to the mainframe.

THE EMULATOR ON THE NETWORK

As part of a LAN, a terminal-emulation package can serve individual users who need to access the mainframe. The only concern is that the emulator settings should not conflict with any other devices within the PC workstation.

For example, if you have a modem in a PC, a 3270 emulation board and a network interface card, you must know the interrupts and all the I/O (input/ouput) channels being used by each. If some of the interrupt settings are identical for two or more boards, the system would not function properly.

Most emulators permit the concurrent use of other communication software, including networking software, as long as the interrupts and channels are not duplicated. The emulator software should permit the changing of channels so that conflicts can be avoided. If a specific address is already used for communication, you should be able to reset the emulator address to an alternative address so that the communication programmes do not "collide" with each other.

The network sets up its own communication system, managed by the network interface card and network software. The network does not affect the PCs' three parallel ports (LPT1, LPT2 and LPT3) and two serial ports (COM1 and COM2). But other devices, such as printers, modems and emulators, use these ports and must be directed properly.

MAINFRAME GATEWAYS: ALTERNATIVES TO TERMINAL EMULATION

The mainframe gateway is an alternative to the terminal emulation package. Instead of serving an individual PC as the standard emulation package does, the mainframe gateway can serve all PCs attached to the network. Although gateways are available that support such protocols as X.25 and asynchronous communication,the SNA gateway is becoming the primary LAN-to-mainframe link.

In most configurations, the gateway emulates a 3274 cluster controller (see Figure 11.4). PCs attached to the LAN can initiate a 3270 session with the mainframe through the gateway. To start the session, the PC must load an emulation programme; the PC requires no special emulation hardware.

Gateways have a fixed number of ports, each of which supports a mainframe session. Gateways come in 8-, 16-, 24-, 32-and 64-port versions. One of the gateway's advantages is that it usually supports as many as 5 or 6 simultaneous mainframe sessions on a single PC. Each session takes up the entire screen, and users can switch from one session to another.

Functionally, the gateway and the emulation board solutions are basically the same. Both let the PC function alternately as a 3278 terminal and a microcomputer. The primary reason for opting for the gateway-type link is a significant saving in cost.

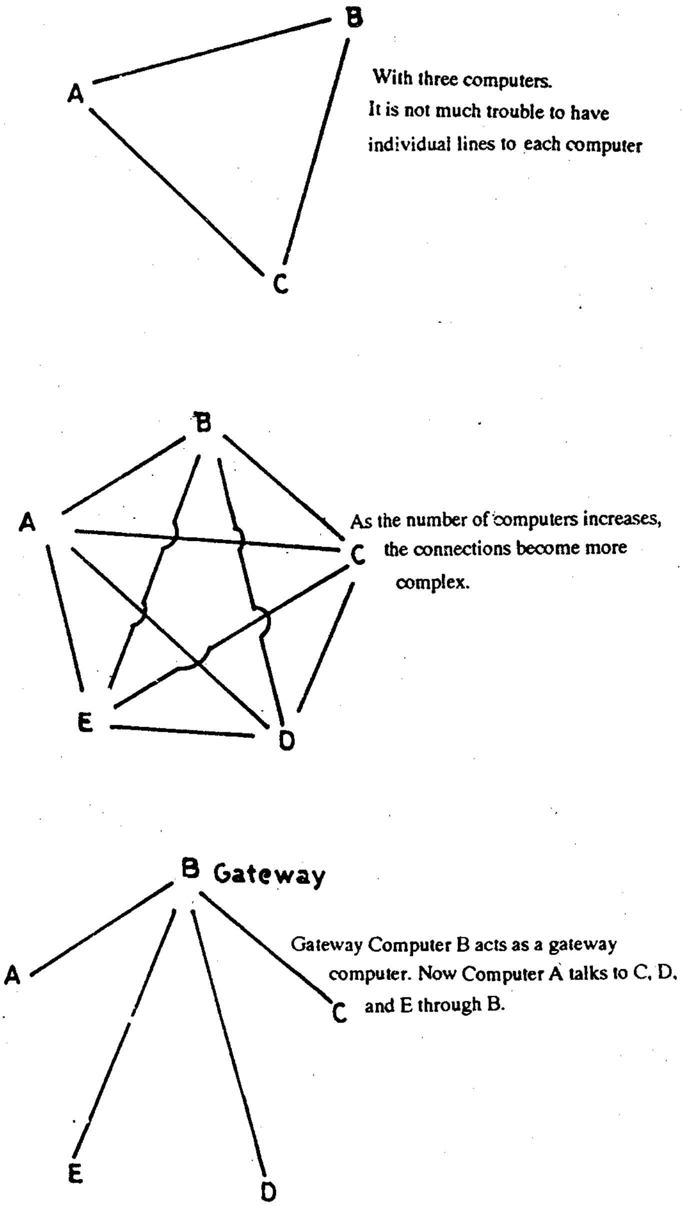

Fig. 11.4 : Reason for gateway

A port for a PC on an SNA gateway costs upto $400. Providing the same connectivity with an emulation board costs approximately $1,000 for the emulation board and another $500 or so for a port on the 3274 cluster controller.

Additionally, the gateway approach requires only one connection between the

LAN and mainframe. With emulation boards, each emulator-equipped PC must be furnished with its own dedicated 3270 cable, usually at a cost of several hundred dollars each.

The resulting cost difference between the two solutions is considerable. Compared to the emulation board solution, the gateway method saves well over $1,000 per PC workstation.

Another important advantage of the gateway is that it permits better management of communication and mainframe processing resources. Emulator-equipped PCs can place a heavy burden on a mainframe communication system because of their capability to download large files, run multiple sessions and use mainframe storage areas through virtual host storage (VHS).

A gateway enables many users to access the mainframe. But the number of simultaneous sessions from the LAN to the mainframe can be limited by restricting the total number of gateway ports available. A 15-workstation departmental LAN, for example, may have an 8-port gateway, which reduces the potential load on the mainframe system. As the network grows, and needs additional mainframe ports, more gateways can be added.

PCS AS MAINFRAME WORKSTATIONS

For an analyst or for anyone else whose position requires that many jobs be done concurrently, the terminal-emulating PC can be effective. Perhaps an analyst works on the mainframe, writing and maintaining programmes. In addition, the analyst may be needed to support local PC users and their application software. With a LAN-attached PC that can emulate a mainframe terminal, the analyst can work in both the PC and mainframe environments and reach out to many physical locations — all from a single workstation.

Suppose that the analyst is working on a project on the mainframe in a 3270 mode and somebody comes in with a question about a spreadsheet. The analyst presses a couple of keys and brings up the application. After assisting the person, the analyst returns the mainframe to 3270 mode. With electronic mail this process is even better. A question can be asked through a PC-conveyed memo. The sender references the appropriate files, which the analyst can call up from the central hard disk. The analyst can then send back a response. Although the two users may be miles apart, the interaction is the same as if they were sitting next to one another.

Another interesting feature of the PC and terminal emulator is that you can capture a complete session. Later, at the end of the session, you can decide whether to keep it. But at the end of a session on a dumb terminal, the session is gone. If you want to know what you did or you want to remember a particular piece of information, you have to run the session again from the start. With the PC, you capture the session and review it as you wish. And you can store the whole session for later viewing or analysis.

Many terminal operators — the people in planning, finance, accounting and inventory — extract bits and pieces from large files. Typically, these operators look through thousands of pages of output but actually manipulate just a few pages of it. With a 3270 emulator, the operators are able to extract information and manipulate it locally on their PCs. The small computational work is taken down to the local level. This set-up reduces the workload on the mainframe, which can then support more operators and be more accessible to them.

With this arrangement the host system is benefited by downloading menial tasks that do not require mainframe power. The mainframe is best used to store very large files and databases. Of course, the mainframe excels at running large programmes and handling extensive computational tasks, such as executing large compiling jobs. All other work should be run on the PC, if possible.

At the local level, operators can manipulate the data with personal computer programmes. These are usually much easier to use than comparable mainframe programmes. And PC software is not only easier to use but often more powerful as well. Usually, mainframe programmes lack the sophistication of the smaller programmes that are written for the PC.

Engineers can bring down subroutines from their files and manipulate these subroutines on the text editor, may be even run a compatible compiler on the PC level. When a subroutine checks out, an engineer can put it into the mainframe and do the final run that costs many thousands of rupees per hour of computation time. The user saves time in manipulating the programme, which, of course, saves money too.

Sometimes, part of the processing is done on the host mainframe, and another part is downloaded to the PC for processing. Suppose that you have a computer programme which generates all its numbers in English engineering units, such as pounds, feet, inches, and so on. You are doing work for a government agency that has decided to convert to metric. The agency wants the report printed in metric units.

You have several choices. You can do all the calculation by hand. Or you can modify your primary programme that generated the numbers. You can leave the primary programme alone and run a conversion programme on the mainframe to put the file into proper form. Or you can let the PC handle the work. You can capture the output onto your PC and write a conversion programme that reads that output, then gives the metric equivalents. Using the PC in this way is good management of resources. You can take data in its original form and have it manipulated for a special purpose by your local machine. You can reformat data and merge it into a document or spreadsheet, all done locally on your PC.

BACK-UP ON THE PC

In a corporate data processing environment, the mainframe computing system provides the primary data storage facility. But even with all the mainframe's sophisticated data protection and back-up procedures, sole reliance on this system has some drawbacks. Mainframe hard disk devices are reliable but not immune to head crashes. Nothing is quite as horrifying as to find out that a head crash occurred on the track where your data happened to be located. Of course, everything is archived or backed up regularly. But suppose that the major editing which you did yesterday was wiped out by a head crash, shortly before the data was due to be backed up.

Besides head crashes, mainframe data is subject to other threats. Disk space on mainframes is always a scarce commodity. Invariably, the amount of data expands to fill the disk faster than new disks can be installed. In one instance, a company moved its computing resources from one computer to another. By mistake, all the archived tapes from the old machine were placed on scratch status, which means that they could be erased and reused. Consequently, vast amounts of important data were lost.

After the arrival of PC, with a handful of floppies, a user have both control and back-up and not be concerned with how the mainframe data is handled or mishandled. Obviously, personal archiving can get you into trouble. You should always work with the programme that is on the mainframe, not with your personal back-up. Otherwise, local back-up systems create the risk of having multiple versions overwriting other people's work, and experiencing many of the multiuser problems. Then, too, personal archiving may violate company security. Nonetheless, a system of personal archiving, when properly used, can be an important benefit of the PC-to-mainframe connection.

RELIABILITY OF MAINFRAME CONNECTION

Many multibillion dollar corporations have trouble maintaining reliable communication between terminals and host mainframe computers. Downtime may have any number of causes. In one instance, a backhoe operator dug through a cable and shut down a data processing department for two days. Most problems tend to be less radical, involving only a few hours of downtime now and again. But accumulated downtime is a serious mainframe-related problem. One study in a major corporation showed that the host computer was unavailable for 10 to 20% of the time. A more representative figure may be as low as 5%. That 5%, however, seems to be concentrated during peak hours, therefore magnifying the effect.

The costs of downtime can mount up with an unreliable communication network. Companies often do not see these costs because they do not show up on any ledger sheet. Nevertheless, the costs are real.

In contrast, the PC rarely has downtime, certainly nothing like that of mainframe systems. If the mainframe goes down, you will probably have an entire staff sitting around with little to do. But if you have copies of mainframe files and programmes sitting on a network server, the real effect of mainframe downtime can be minimised.

CONNECTING MAINFRAMES TO LANS

In 1986, IBM released dozens of software products for their local area networks. The networks themselves provided connections between devices; the software products continue the process by providing connections between applications.

In this section, we will discuss the products available for connecting each kind of mainframe to each kind of LAN. Keep in mind that some of the products are not sufficient for the average user's needs. What exist are the physical hardware connections and some of the general software connections. But you cannot yet run Lotus 1-2-3, for example, and get data directly from the mainframe.

In general, IBM's Token-Ring Network is a better bet than the PC Network if you plan to communicate with mainframes. For example, the Token Ring Supports two programming interfaces: NETBIOS and APPC. Although not much software is available to take advantage of APPC, the fact that it is supported by the Token Ring is, as IBM would say, a strategic statement of direction. NETBIOS does not support the LU 6.2 protocol at all. Consequently, Token-Ring Network users must choose between running NETBIOS to be compatible with existing PC networks, or APPC to be compatible with future mainframe communication.

MAINFRAME PRODUCTS FOR THE TOKEN RING

Communication products for the Token Ring are divided into three categories, depending on which mainframe is being used:

- System/370 and 370 compatibles
- System/36
- Series/1

We will discuss each computer family in turn, first covering the hardware products available, then the software.

System/370 Hardware for the Token Ring

The System/370 is IBM's flagship mainframe computer. Computers in this family are more popular than any other mainframe computer. They are most often used for administrative duties, such as budgeting, storing personnel records and managing large databases.

Terminals and computers communicate with System/370 machines through hardware devices called communication controllers. These devices are like the control units on expensive stereos. Communication controllers have connections for devices such as terminals, networks, consoles and modems. Most of the ways to communicate with a System/370 involve the 3725 Communication Controller. In addition, PCs can communicate with the System/370 indirectly through the System/36; this is discussed in the section on System/36 communication.

3725 Communication Controller

The 3725 Communication Controller has been around for several years. In general, it is used to communicate between terminals and the mainframe. With a special attachment, up to four Token-Ring Networks can be connected to a Model 1 or Model 2 3725; by adding an expansion unit called a 3726, a total of eight Token-Ring Networks can be connected.

To connect the LANs to the 3725, you first need a unit called a Line and Token-Ring Attachment Base [LAB] Type C. The LAB has a twofold purpose:

- to hold the TICs [Token-Ring interface couplers - these are what actually communicate with the Token-Ring Network by transmitting and receiving messages].
- to funnel the information from the TICs to the mainframe.

Each LAB unit can contain four TICs; this means the LAB can communicate with up to four Token-Ring Networks. Each TIC can connect to one Token-Ring Network with a shielded twisted-pair [type 6] cable. A 3726 expansion unit can be connected to the Model 1 3725 controller. Combined, the 3725 and 3726 can contain two LAB Type C units; that is, the 3726 can hold two if the 3725 has none, or the 3725 and 3726 can each hold one. The TIC can transmit data at rates of up to four million bits per second. Speed in actual use, however, depends on factors such as system load.

3720 Communication Controller

The 3720 is a small communication controller in the 3725 family. Like the 3725, it is used to communicate between terminals and the mainframe.

Four 3720 models are available; the ones we are concerned with are the Models 11 and 12, which can each support upto two Token-Ring Networks. The Model 12 is designed for remote locations connected to the mainframe by telephone lines. They do this through the same TIC connector that the 3725 uses; LABs are not needed. Each can also communicate with upto four System/370 computers through connections known as TPSs for two-processor switches. As the name implies, one switch can connect two mainframes, and the 3720 can support two switches. Like the network adapter cards used to connect PCs to the LAN, the TICs and TPSs are optional cards that are inserted in the 3720.

Performance of 3720 is supposed to be about one-third that of the 3725's;consequently, one might expect the maximum transmission speed to be slightly over one nillion bits per second. As with the 3725 in-use speed depends on system load.

3174 Subsystem Control Unit

The 3174 Subsystem Control Unit connects 3270-type terminals to System/370 mainframes through a 3720 or 3725 Communication Controller. Because of the 3270-type terminal's high intelligence, the 3174 is required to manage data transfer between the terminal and the System/370. Through the 3174, PCs on a Token-Ring Network can also communicate with the high-priced, high-performance 308X and 3090 mainframes.

There are two varieties of the 3174, the large cluster and the small cluster, with several models of each variety. Token-Ring Networks can be connected to one model of each variety. The large cluster model that supports the Token-Ring Network is called the 3R, and supports up to 32 terminals. The small cluster model that supports the Token-Ring Network is called the 53R, and supports up to 16 terminals.

3270 Emulation

The final hardware communication method to a System/370 is also the oldest: terminal emulation. One PC emulates a different type of communication controller, the 3274, while up to 32 other PCs emulate 3278 [monochrome] or 3279 [colour] terminals. It is called 3270 emulation because all these devices are members of the 3270 family.

Although this method has the advantages of simplicity and low cost, it has all the inherent disadvantages of terminal emulation. Moreover, the PC emulating the 3274 controller must be dedicated to this task; it cannot be used for any function other thanas a terminal controller. If one PC is emulating the 3274 terminal controller, a card called an SDLC adapter is installed in that PC. This card is similar to the network adapter card.

System/370 Software for the Token Ring

The hardware connection between the System/370 mainframe and the Token-Ring Network is only half the story. To do useful work, there must be a software connection as well. Three kinds of software can connect Token-Ring Networks and mainframes; most of the software supports several kinds of hardware connection. In addition, the communication controllers usually have some supporting software. Support for the Token-Ring Network is usually included in the latest release of the software, so selecting the proper software is not an issue.

APPC Software

No APPC software exists as yet. Nonetheless, it is almost certain that software using the APPC standard will be developed both by IBM and by other vendors. Software written to this standard will be supported on the 3174, 3720 and 3725 hardware connections.

Personal Services/PC Software

You use Personal Services/PC software when you are communicating with the previously described DISOSS on the System/370. With Personal Services/PC, you can access and manipulate documents from a Token-Ring Network just as you can when you are using a terminal on the System/370. For example, you can store, retrieve and delete documents in the System/370 host library. You can also search for particular documents, and then print them on printers attached to the mainframe. Finally, you can send documents to other users with the electronic mail portion of Personal Services/PC.

Most models of the IBM PC support Personal Services/PC. Unfortunately, Personal Services/PC for the System/370 is supported only when the PC is emulating a 3270 terminal. This means that the programme cannot take full advantage of the

PC's intelligence, for reasons we mentioned previously when discussing terminal emulation. The Personal Services/PC software can be used when the PC is connected with a 3174, 3270, 3725 or 3720 hardware controller device, as long as the PC is emulating a 3270 terminal.

3270 Emulation Programme

The IBM PC 3270 Emulation programme is used when PCs connected to the Token-Ring Network need to emulate 3270 terminals to communicate with the System/370. Several versions exist; all versions of the software let users run System/370 programmes by emulating a 3270 terminal.

This software is used when a PC is emulating a 3270 terminal, either when a PC is directly connected to a 3274 terminal controller or when a PC is emulating a 3274 terminal controller. In addition, PCs connected to either a 3725 or 3720 Communication Controller can run this software, rather than APPC, if you need or prefer terminal emulation - for example, if you want to run Personal Services/PC.

System/36 Hardware for the Token Ring

The System/36 is a small computer that IBM calls a "departmental computer"; it can satisfy the computing needs of a department, whereas the System/370 can satisfy the needs of an entire company or a division. Several models of the System/36 are available. IBM has called the System/36 the primary office departmental system; the machine's support of the Token Ring bolsters this.

5360/5362/5364 System Unit LAN Attachment

The 5360, 5362 and 5364 are all different models of the System/36 computer. The 5360/5362/5364 System Unit LAN Attachment devices connect the Token-Ring Network to System/36 machines. In the case of the Token-Ring Network, the connection is through a dedicated IBM PC/AT — that is, the only function the PC/AT can perform is that of a communication controller. When connected to the 5364 model of the System/36, the IBM PC AT can run System/36 workstation applications, but cannot run IBM PC programme. A System/36 can communicate with upto two Token-Ring Networks.

With the 5360 and 5362 models of the System/36, an additional device called the System/36 LAN Attachment Feature is required. This product includes a System/36 LAN Attachment Adapter card, which is installed into the IBM PC/AT like any typical card. The 5364 can communicate directly with the dedicated IBM PC/AT, so the System/36 LAN Attachment Feature and LAN Attachment Adapter card are not required.

3174 Subsystem Control Unit

With the 3174 Subsystem Control Unit, you can connect a Token-Ring Network directly to the System/36 mainframe. As with the System/370, the 3174 connects 3270-type terminals to the System/36. Because of the 3270-type terminal's high intelligence, the 3174 is required to manage data transfer between terminals and the System/36.

There are two varieties of the 3174, the large cluster and the small cluster, with several models of each variety. Token-Ring Networks can be connected to one model of each variety. The large cluster model that supports the Token-Ring Network is called the 3R, and supports up to 32 terminals. The small cluster model that supports the Token-Ring Network is called the 53R, and supports up to 16 terminals.

PC Adapter II for the IBM PC

A PC Adapter II card must be installed in the dedicated IBM/PC AT for each Token-Ring Network connection to the System/36. That is, if the maximum of two Token-Ring Networks are connected to the System/36, two PC Adapter II cards are required. This is true whether the 5360, 5362 or 5364 models of the System/36 are used.

The PC Adapter II card can also be used to connect IBM PCs to the Token-Ring Network; it is the equivalent of the PC Network adapter card with an additional 8K of RAM. The additional RAM lets you send longer network messages and, when installed in a server, support more PCs.

The PC Adapter II card can be installed in the IBM PC, Portable, PC/XT, PC/AT and models 5531, 7531 and 7532 of the Industrial Computer. To act as a controller to the System/36, however, an IBM PC/AT is required. As with the PC adapter card, the PC Adapter II supports both the APPC and NETBIOS programming interfaces.

System/36 Software for the Token Ring

Three kinds of software are involved in the System/36 connection to the Token-Ring Network. Two of them are sold together under the name LAN Communications Licensed programme; one component runs on the System/36 and one component runs on the IBM PC/AT used as a gateway between the System/36 and the Token-Ring Network. The third kind of software, Support/36, runs on the PCs communicating to the System/36.

System Unit LAN Communications Licensed Programme

The System Unit LAN Communications Licensed programme comes in two versions: one for the 5630 and 5632 models of the System/36, and one for the 5634 model. You do not run this programme, as such; it just has to be available on the system unit. The programme supports communication with upto two Token-Ring Networks.

The component that runs on the gateway IBM PC/AT is stored on the System/36. When communication start, the System/36 downloads this component to the IBM PC/AT, where the programme automatically runs.

Support/36 Programme

The Support/36 programme is what lets PCs on the Token-Ring Network communicate with the System/36. The optional Support/36 Workstation Feature lets the PC emulate a System/36 workstation as well. With the Workstation Feature, each PC can emulate up to four System/36 workstations; although only one System/36 session can be displayed on the screen at a time, any other sessions continue running. However, a PC session cannot continue running. One of these sessions can emulate a 5219 or a 5256 printer.

Even without the Workstation Feature, PCs can use the System/36 as a virtual disk (disk server) and virtual print (print server) device. Essentially, the programme gives any PC on the Token-Ring Network the same capabilities as a directly connected mainframe PC. With this software, PCs can also emulate a 3270 terminal and can communicate through the System/36 to a System/370.

Series/1 Hardware for the Token Ring

No products, specifically for the Token-Ring Network, are available to communicate with the Series/1. Several products that can run on the PC Network, however, can run on the Token-Ring Network tooif the Token Ring is running NETBIOS. See the following descriptions.

MAINFRAME PRODUCTS FOR THE PC NETWORK

In general, fewer products are available for the PC Network than for the Token-Ring Network. For example, the PC Network has no way to communicate with the System/36. Although some products are available, it is better to buy a Token-Ring Network if you need to do large-scale mainframe communication. All products mentioned in this section will also run on the Token-Ring Network if it is running NETBIOS.

System/370 Hardware for the PC Network

PCs on the PC Network can communicate with a System/370 only through terminal emulation. Terminal emulation is an inefficient way to communicate between PCs and mainframes. For this reason, IBM is recommending that LAN users, who often need to communicate with a mainframe, use the Token-Ring Network.

3270 Emulation

The only communication method from the PC Network to a System/370 is also the oldest: terminal emulation. It can be accomplished in one of two ways. First, one PC can emulate a different type of communication controller, the 3274, while upto five other PCs emulate 3278 (monochrome) or 3279 (colour) terminals. It is called 3270 emulation because these devices are all members of the 3270 family. Second, a PC can be connected directly to a 3274 terminal controller.

Although these methods have several advantages, namely their simplicity and low cost, they all have the inherent disadvantages of terminal emulation. Moreover, the PC emulating the 3274 controller must be dedicated to this task; that is, it can not be used for any function other than acting as a terminal controller. In a PC emulating the 3274, you install a card called an SDLC adapter. This card is similar to the network adapter card.

System/370 Software for the PC Network

3270 Emulation Programme

There are several versions of the IBM PC 3270 Emulation programme. All versions of the software let users run System/370 programmes by emulating a 3270 terminal. In addition to running System/370 programmes, you can run Personal Services/PC to access DISOSS features.

Personal Services/PC Software

You use Personal Services/PC software when you are communicating with DISOSS on the System/370. With this product, you can access and manipulate documents just as you can when you are using a terminal on the System/370. For example, you can store, retrieveand delete documents in the host library. You can also search for particular documents, and then print them on printers attached to the mainframe. Finally, you can send documents to other users with electronic mail. Most models of the IBM PC support Personal Services/PC. Because communication to the System/370 is supported only when the PC is emulating a 3270 terminal, Personal Services/PC can not take full advantage of the PC's intelligence.

Series/1 Hardware for the PC Network

The traditional role of the Series/1 has been that of a factory processor - controlling robots, industrial manufacturing machines, and so on. IBM has begun adding office functions to the Series/1, to the extent that some analysts think it will eventually replace the System/36. But the fact that the Series/1 does not support the Token-Ring Network (other than in NETBIOS emulation) might be an IBM signal that it will not be enhanced much further.

Series/1 PC Connect Adapter

As with the System/370 connection, you can connect to the Series/1 by dedicating an IBM PC/AT or XT for use as a gateway to the PC Network. Then, any PC on the PC Network can communicate with the Series/1 mainframe, including using its disks and output devices. The adapter is much like any other PC card, and you install it the same way.

Series/1 Software for the PC Network

As mentioned, IBM recently began adding office functions to the Series/1. For example, the machine now supports the DISOSS protocols. Consequently, PCs on the PC Network can access documents on the Series/1 with the Personal Services/PC package. PCs connected to the Series/1 mainframe can communicate with System/370 DISOSS, but because information is transferred using SNADS and other IBM data communication connections, rather than with the PC Network or the Token-Ring Network, it will not be discussed here.

Series/1 Office Connect Software

Office connect includes software for the PC as well as the Series/1. With Series/1 software, the machine can provide the document distribution and library services. With PC software, PCs can communicate with Series/1 software using Personal Services/PC (described next).

Personal Services/PC Software

Personal Services/PC software is used to communicate with Office Connect on the Series/1. With it, you can access and manipulate documents just as you can when you are using a terminal on the System/370. For example, you can store, retrieve and delete documents, and then print them on printers attached to the mainframe. Finally, you can send documents to other users with electronic mail. Most models of the IBM PC support Personal Services/PC.

FUTURE OF MAINFRAME AND LAN COMMUNICATION

By now, you may have come to two conclusions. First, mainframe to LAN communication is complicated. Second, it will be a while before all the pieces are in place for complete mainframe-to-PC networking.

Although a base for peer-to-peer communication is there, you are still limited to terminal emulation with existing products. But by the end of the decade, you should have your choice of products that support peer-to-peer communication between mainframes and LANs. Now we will discuss the future of mainframe and LAN communication, including problems that must be solved and where the first software products will come from.

What does peer-to-peer communication really mean? How will it affect your life? What good will it do you? Let usstart with an example. Suppose you want to get some of your company's budget information out of the mainframe and into a Lotus 1-2-3 spreadsheet. Most likely, you cannot do it directly (and if you can, you can bet that somebody in the MIS department spent a few busy weeks writing a programme to enable you to do it).

So how would you do it? First of all, you have to run a programme on the mainframe to extract the data you want from the budget, and place that information in a separate file on the mainframe. Second, you have to transfer the information from the mainframe to the PC, which probably requires running a terminal emulation programmeand either downloading the file or logging it to disk. Finally, you have to convert the mainframe data into a form that Lotus 1-2-3 on the PC can read and that is in the same format as your spreadsheet.

By now you have probably concluded that it would be easier to retype the spreadsheet information. But let us look at how this scenario might work with APPC-compatible applications. First, you run Lotus 1-2-3 and call up your spreadsheet. Next, while you are still running Lotus 1-2-3 with the spreadsheet, you look at the budget data on the mainframe. It would automatically be converted to a format that Lotus 1-2-3 and your spreadsheet could read. Finally, you select the data you need and it is automatically transferred to your floppy disk.

When you have finished, the software might automatically update the information in the mainframe's files. Or you might want to check it first; in that case, the software could save a copy of the file on the mainframe to protect you from power and disk problems.

But what if your company's budget was so large or so disjointed that it was spread over several computers. In that case, the software might find out which computer contains the information, set up a session on that computer, and transfer the information. All of this would happen without your doing anything, or even without your noticing. This feature is called transparent access, because you cannot see it happening. APPC has other potential advantages. Currently, most information has to pass through a mainframe computer. In the office product protocols, all distribution goes through a central computer.

But what happens if that computer fails? Anybody who has ever needed to fly into a hub city that is having problems knows what happens - air traffic backs up all over the country. The same thing happens with the computer network. In the future, however, things will be different. Computers will know several routes for transmitting information, so if a certain route goes down or is overloaded, data can simply be transmitted by another route.

Vendors of computers other than IBM, such as Apple and DEC, are looking at APPC too. With it, computers that are not compatible with IBM, such as the Macintosh, could transmit data to IBM PCs without having to be fully IBM-compatible.

This intercomputer communication may require time. Fortunately, APPC supports multitasking (doing several things at one time). Consequently, if it is taking a long time to get your data, you can continue writing your report or revising your spreadsheet. Perhaps a window in the corner of your PC screen will inform you of the progress of the data transfer.

These examples are just a sample of what APPC-compatible programmes might be able to do. Probably the first programmes to support the APPC interface will be new products, or products that are already popular, such as Lotus 1-2-3. Eventually, products that do not support the APPC interface will be ignored, as tapes in Betamax format are now.

POTENTIAL ROADBLOCKS TO PEER-TO-PEER COMMUNICATION

Before this idyllic scenario can take place, several issues need to be settled. First, IBM has not finished the final APPC specifications yet. This definitely slows development; you can not write a programme to fit a standard if the standard itself has not been defined.

Second, it will be many years before mainframe applications can be converted to peer-to-peer applications, because so many mainframe programmes are still written in COBOL and assembly language. MIS departments can not rewrite programmes fast enough. And until those programmes are converted, users will be stuck with terminal emulation and master-slave communication.

Then there is the question of how users will manage all the data that is suddenly available. Any organisation with more than one PC and more than one copy of Lotus 1-2-3 has probably experienced the multiple spreadsheets problem. One of these days, a scientist will win the Nobel Prize by explaining how several users, starting with the same data, can reach different conclusions - some of them flawed. And the more access users have to data, the more faulty conclusions they can come up with.

Worse still, users could take data produced by their faulty conclusions and transmit it to the mainframe, where it would replace the correct data. Organisations need to set up procedures and guidelines to prevent this.

Transactions also need to be defined so one user does not rewrite a record just as another user is reading it. Until these data management issues are settled, application programmers have to incorporate the precautions into their programmes. But if each programmer devises a different way to incorporate data management, we could again have incompatible programmes.

Then there is network management. Everyone certainly has heard of the Postal Service's Dead Letter Office, with its storehouse of letters that are addressed to unknown places or have illegible addresses. In the same way, if a computer does not know how to reach the destination computer, the message never gets to the recipient. Whenever computers are removed from, added to or moved within a network, their addresses change and there is chance for messages to be lost.

Keeping track of computer whereabouts is already a problem in traditional SNA networks, which are set up so that most routing information is contained in a few computers. This is equivalent to having all your letters go to New Delhi before they can go to their destination across town. Every time a computer is added to or removed from the network, for example, the entire network has to shut down to change the address tables.

A peer-to-peer environment would be worse, because it cannot be centrally configured. Instead, each computer must know the addresses of all the other computers, including how to get there. It is as though a cross-country letter has to include the names of all the cities through which the letter might travel.

This information grows geometrically. Every time a computer is added to the network, it must be configured with the addressing information about all the other computers in the network.

Fortunately, IBM is developing an extension to SNA called LEN, for Low Entry Networking. When this extension is fully developed, network configuration will be easier. Moreover, it may implement dynamic routing, which means that the route from one computer to another can change depending on network conditions. If traffic is slow or a computer is down, for example, the route can be changed to handle the situation.

This brings up another problem, however. When there is no central computer, how do you know when a computer or a communication line is not working? The malfunctioning computer cannot tell you. Existing central computers can run diagnostics on other computers, but with APPC, there will not be any central computers.

After enough messages are lost, someone may finally realise that there is a problem, but that could take days. It might be like the movie, the Andromeda Strain, in which the scientists did not receive any important messages for several days simply because a piece of paper was blocking the bell on the teletype machine ! IBM and third-party vendors need to develop diagnostic hardware and software that can deal with LANs.