Inside the Token-Ring

3rd Edition

J. Scott Haugdahl

ARCHITECTURE
TECHNOLOGY
CORPORATION
SPECIALISTS IN COMPUTER ARCHITECTURE

Haugdahl Inside the Token-Ring 3rd Edition Architecture Technology Corporation

Inside the Token-Ring

3rd Edition

J. Scott Haugdahl

ARCHITECTURE
TECHNOLOGY
CORPORATION
SPECIALISTS IN COMPUTER ARCHITECTURE
P.O. BOX 24344 • MINNEAPOLIS, MINNESOTA 55424 • (612) 935-2035

Printed in the United States of America

First Printing April 1986
2nd Edition Printing January 1988
3rd Edition Printing May 1990

Published by:
Architecture Technology Corporation
P.O. Box 24344
Minneapolis, MN 55424

Library of Congress Cataloging-in-Publication Data

Communication Networks.

Includes Index.
1. Computer Networks 2. SNA
I. Haugdahl, J. Scott

ISBN 0-939405-00-8

DISCLAIMER

FOREWORD

When IBM announced its Token-Ring in October of 1985, there was much skepticism and criticism surrounding this new IBM LAN -- after all, it could only connect PCs and the software was no match for established PC LAN vendors. Now, short of five years later, the Token-Ring provides connectivity for every major computing device that IBM offers in addition to several non-IBM offerings. In addition, it is the strategic local area network included in IBM's Systems Application Architecture (SAA).

The Token-Ring has achieved status in the local area network marketplace that took Ethernet ten years to achieve (Ethernet is now in its sixteenth year). The blitz of products that IBM has announced over the past four years are shipping or about to ship, with more announcements to come. Token-Ring purchases of 10,000+ nodes at a time are not unheard of. Needless to say, the critics are somewhat embarrassed by the success of the IBM Token-Ring. While Ethernet continues to currently dominate, we can not ignore this "new" technology.

This book covers all aspects of IBM's Token-Ring local area network, from history to IEEE Standards to interfaces and products. The reader of this report will gain a solid understanding of the IBM cabling system, token ring operation, and management features unique to the token ring.

J. Scott Haugdahl
Minneapolis, Minnesota
May, 1990

TABLE OF CONTENTS

List of Figures

To Nancy and Daniel

Chapter 1

Overview

History

The origins of token-passing on a ring is subject to debate, but one of the older ring control techniques dates back to 1969 to E.E. Newhall, after whom the Newhall Ring was dubbed, and D.J. Farber's Distributed Computing System at the University of California at Irvine in 1972. While the Cambridge ring/slotted ring technique was the prevalent ring control strategy in England and Europe, it has been surpassed by the token-passing on a ring technique which is also popular in the U.S. This popularity can be attributable in large part to IBM's long-stated commitment to the token-ring method -- and the fact that it is the one ring access method selected for standardization by the IEEE 802 Local Network Standards Committee, in its 802.5 token-ring specification.

IBM's first public inklings with token-rings was embodied in a series of four papers IBM presented to IEEE 802 in March of 1982 and at a conference (IFIP) in Florence, Italy, in April 1982. The IEEE 802 presentation described a new architecture; the Florence presentation described some implementations of that architecture. In August of that same year, at an IBM users' group meeting, IBM presented a fifth paper on wiring. IBM then took the IEEE 802 papers and presented them at the IEEE Computer Society's semi-annual conference, COMPCON fall '82.

The next major development was a public demonstration of a token-ring prototype at Telecom '83 in Geneva, Switzerland, October 26 through November 1, 1983. This demonstration was a simplified version of today's token-ring. An attached device gained access to the network by changing the status of a perpetually circulating 1-bit token from "free" to "busy;" the token is in the header of a message frame, and this frame is then filled with all or part of the message itself. The demonstration ring consisted of a wiring concentrator connected to several terminals -- 3270 terminals, 8775 display terminals, a 3275 front-end processor, and 8100 distributed processors.

In May of 1984, IBM announced the IBM Cabling System that was designed to connect all IBM data communications devices and accommodate the yet-unannounced token-ring.

In 1985, the token-ring became an ANSI/IEEE standard. IBM officially announced the Token-Ring in October of that year with a series of products for its PC family. The products were developed by the IBM Telecommunications at Raleigh NC, and were an outgrowth of the research performed by IBM Zurich (the infamous "Zurich Ring"). At the time, IBM felt that 4 million bits per second (Mbps) was adequate for departmental/office automation requirements, that the token-passing protocol offers superior (over Ethernet) throughput under heavy loads, and that the protocol is better suited to supporting IBM's synchronous devices.

These product announcements include the Multistation Access Unit (MAU), support for telephone "Type 3" wire, the IBM PC Token-Ring adapter, the Asynchronous Communications Server, the NETBIOS emulation program, enhancements to the 3270 Systems Network Architecture (SNA) Emulation Program, and APPC/PC. IBM stated that the IBM PC Network Program could operate "as-is" but was later released as an upgrade (now called the PC Local Area Network Program), along with PC DOS 3.2.

IBM's initial announcement generated a lot of criticism in the areas of system sizing and interconnectivity. Only a single ring was supported with a maximum of 260 devices over a limited distance. The only direct connection was an adapter for the PC.

Subsequent announcements in April, May, June, and September of 1986 added support for mainframes and the System/36. IBM added token-ring support for the PS/2 when it was announced in April of 1987. In November of 1988, IBM announced several new token-ring products, most notably the 16-Mbps version. The 16-Mbps version was developed in response to Ethernet and also provides a "lower cost alternative" to the 100 Mbps FDDI standard.

Historically, IBM remained aloof from IEEE standards, preferring to create its own proprietary product standards. IBM broke with tradition by involvement in shaping the IEEE 802.5 standard for the token-ring access method. Creation of 802.5 makes the token-ring system open for use with non-IBM products. There are still some problems in areas where IBM has "enhanced" the standard, such as source routing (covered later in this book). These problems are being slowly overcome as IBM has submitted virtually all of these enhancements to 802.5 to become standards.

Another problem area is compatibility -- not all of IBM's token-ring software will operate properly on other vendor's token-ring boards. This is due to differences in adapter interfaces and is not a problem unique to token-ring. Ethernet vendors face the same problem and are working together to form an interface standard (at least for OS/2) called the Network Dependent Interface Specification (NDIS). Unfortunately, as of this writing, there is no such similar adapter interface standard being developed for token-ring. The best hope for token-ring is some type of standard interface to a higher layer of protocol, such as the IEEE 802.2 logical link control (LLC) standard.

Soderblom

In order to become an IEEE standard, the technology must be in the public domain ("open") and all parts must be non-proprietary. Controversy has arisen regarding the Token-Ring, due to the fact IBM holds patents on some of its parts and Olaf Soderblom has patented token-ring LAN technology. IBM has "settled" with Soderblom, paying an estimated one-time fee of five to seven million dollars. Soderblom has apparently waived the need for IBM to pay royalties or fees. Some 40 other companies pay royalties including HP, TI, NCR, Harris, Ungermann-Bass, Proteon, 3Com, Racal, NEC, Fujitsu, and Hitachi.

Another issue of openness is the fact that IBM has enhanced the 802.5 MAC specification to include additional management vectors in the MAC protocol and source routing -- it is the source routing that is somewhat controversial and more difficult for third-party vendors to support than the additional management vectors. Both of these issues are be discussed in more detail later in this chapter.

Cabling System

The IBM Cabling System is the underlying wiring system for the Token-Ring. Key components of the system are: twisted-pair wire, wiring concentrators, connectors, patch panels, and faceplates.

IBM designed the wiring system to be a structured system for all communicating devices, eliminating the previous ad-hoc approach, and, hopefully, help to eliminate future rewiring. Therefore, the cable can be used for more than just token-ring devices -- for example, it can also be used in a point-to-point manner to support connection of 3270 and 5250 terminals to hosts.

The physical topology created by wiring for token-ring resembles interconnected stars, where all devices share the same dual twisted-pair bus. If one were to trace the wire from device to device, a ring is formed eventually wraps back upon itself. Figure 1-1 illustrates this

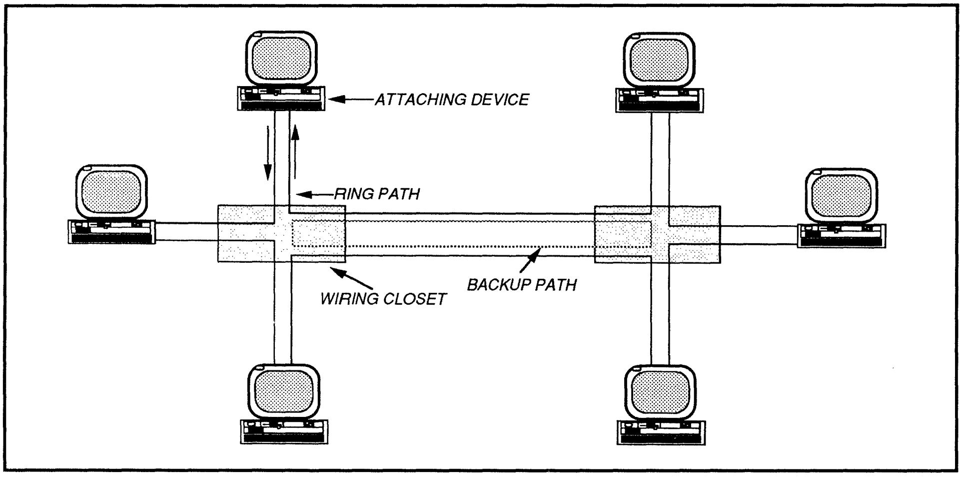

Figure 1-1: Ring Topology

concept.

The Cabling System also supports telephone-type wire (unshielded twisted-pair) for 4-Mbps token-rings. IBM added this support under pressure to compete with AT&T's Premises Distribution System (PDS). The Cabling System is detailed in Chapter 2.

Token-Ring

Token-ring refers to the technique used for controlling access to the ring. Token-ring is a very different protocol than the Carrier Sense Multiple Access (CSMA) protocol used in Ethernet/IEEE 802.3 systems. The IBM Token-Ring protocol is an IEEE 802.5 standard which operates on a ring wiring system. Other token passing protocols, such as Datapoint's ARCNET or the IEEE 802.4 Token-Bus standard for Manufacturing Automation Protocol (MAP) networks, operate on a physical bus.

Access to the token-ring is controlled by a token that circulates from station to station. A station must seize control of the token to send a frame on the ring. This elementary idea is illustrated in Figure 1-2. When the sending station detects the end of the frame that has been passed around and copied by the destination station, it sends the token on to the next station. A variation of this has been introduced by IBM for 16-Mbps operation (and is an 802.5 draft standard). In this case, the token may be released as soon as the station completes the sending of a frame (see Figure 1-3). This early token-release concept al-

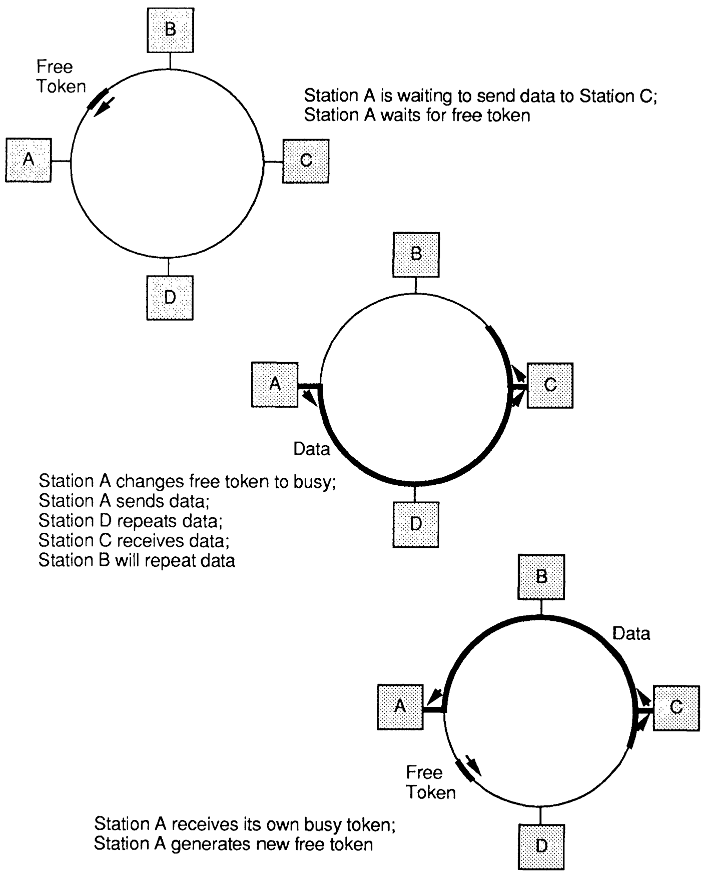

Figure 1-2: Basic Token-Ring Operation

lows for more efficient bandwidth utilization in large rings with lots of stations (with turn-around delay) and/or long cabling (latency in the wire).

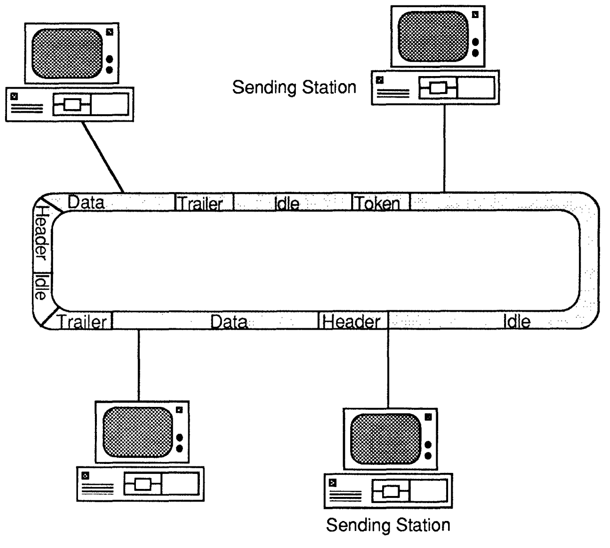

Figure: 1-3: Early Token Release

In reality, the token-ring media access control (MAC) protocol is more complex than the elementary description sounds. A monitor station exists in the ring to ensure proper circulation of the token and frames, and to restrict high-priority token network "hogging." If a station fails, built-in management protocols will cause the failed station to automatically switch out and cause the ring to be re-configured.

The MAC protocol and frame format also includes bits which function as indicators called access control bits. These access control bits tell the sender if a station recognized its address and if it was able to copy it to a buffer. This way, the sender knows if the station is inserted (active) on the ring and if it was able to buffer the frame (i.e the receiving station was not congested in which the sender must retransmit the frame or initiate a flow control algorithm).

Reliability and diagnostics are discussed further in Chapter 7. Details of the protocol and its format are discussed in Chapter 3.

Where the Token-Rings Are

Using a database of over 2000 LAN sites compiled and maintained by Architecture Technology, a survey of over 100 token-ring LANs was randomly selected. The distribution of installed token-rings by industry is given in Figure 1-4. The largest distribution is in services (airlines, insurance, data processing service bureaus, utilities, etc.); manufacturing includes companies such as Corning, LTV Steel, and Toro, but this does not imply that token-rings are on the plant floor!

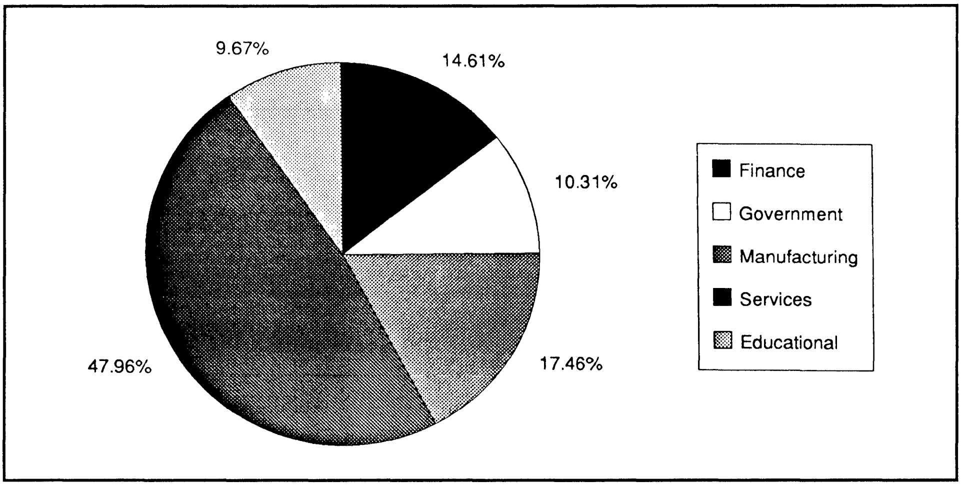

Figure 1-4: Distribution of Token-Ring by Industry

Somewhat surprising was the "low" percentage for the financial category (banks, brokers, etc.). However, when compared with a similar survey for Ethernet, Ethernet comes in at less than 2% in this category. This supports the theory that IBM was able to achieve quick penetration with Token-Ring, since many large "true blue" (in which Token-Ring is the only perceived solution) customers are in the financial sector. In fact, the average number of token-ring nodes is highest in this sector (see Figure 1-5), and the only sector in which the average number of nodes was higher than Ethernet.

Directions and Systems Application Architecture

IBM has stated the Token-Ring LAN will have a 20- to 30-year life cycle. Towards that end, we expect IBM to continue to announce new products and enhance performance.

IBM's strategy for introduction of Token-Ring products has been to bring out products in accordance to "target market" size. In other words, the first are intended for use with the large installed base of IBM PC computers (not to mention compatibles!). This strategy is

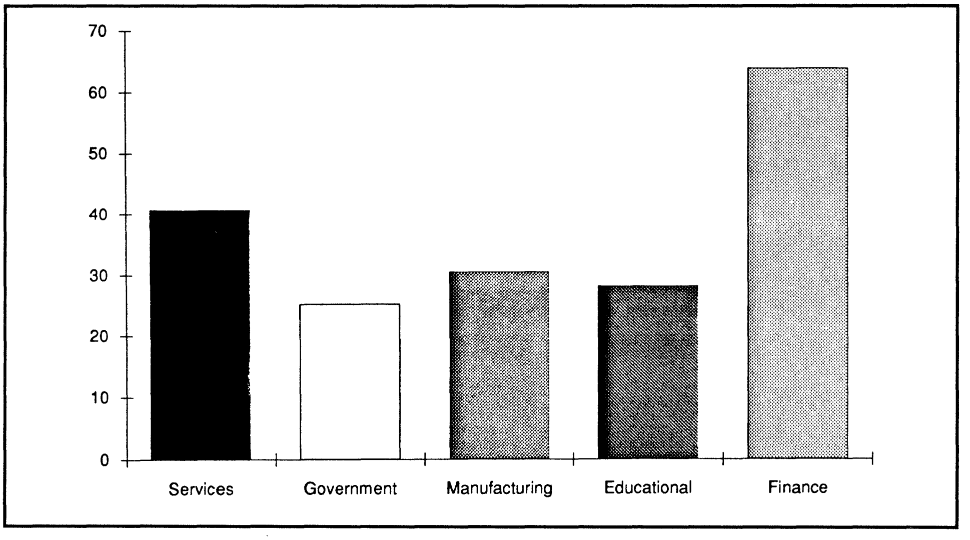

Figure 1-5: Average Number of Token-Rings by Industry

IBM PC computers (not to mention compatibles!). This strategy is paying off for IBM, as it allows IBM to quickly install hundreds of Token-Ring networks in office environments, and several large interconnected (via SNA) networks in corporate environments with PCs and PS/2s as workstations. This strategy is accelerating IBM's foothold in the LAN marketplace, something neither IBM PC Network Broadband nor IBM PC Cluster accomplished and took Ethernet ten years to accomplish. Interestingly, IBM announced PC Network Baseband -- a 2-Mbps StarLAN-like LAN -- at the same time as the PS/2. PC Network baseband was presumably targeted at the educational and small business markets. The positioning of the Token-Ring, PC Network Broadband, and PC Network Baseband is given in Figure 1-6.

On March 17, 1987, IBM announced IBM Systems Application Architecture (SAA), a collection of selected software interfaces, conventions, and protocols published throughout 1987. IBM Systems Application Architecture is the framework for development of consistent applications across the future offerings of the major IBM computing environments -- System/370 (with TSO/E under MVS/XA and CMS under VM), System/3X, and Personal Computer (with Operating System/2). IBM announced the first products to conform to SAA in the Personal Systems/2 announcements in April of 1987 and System/370 enhancement announcements in May and June of 1987.

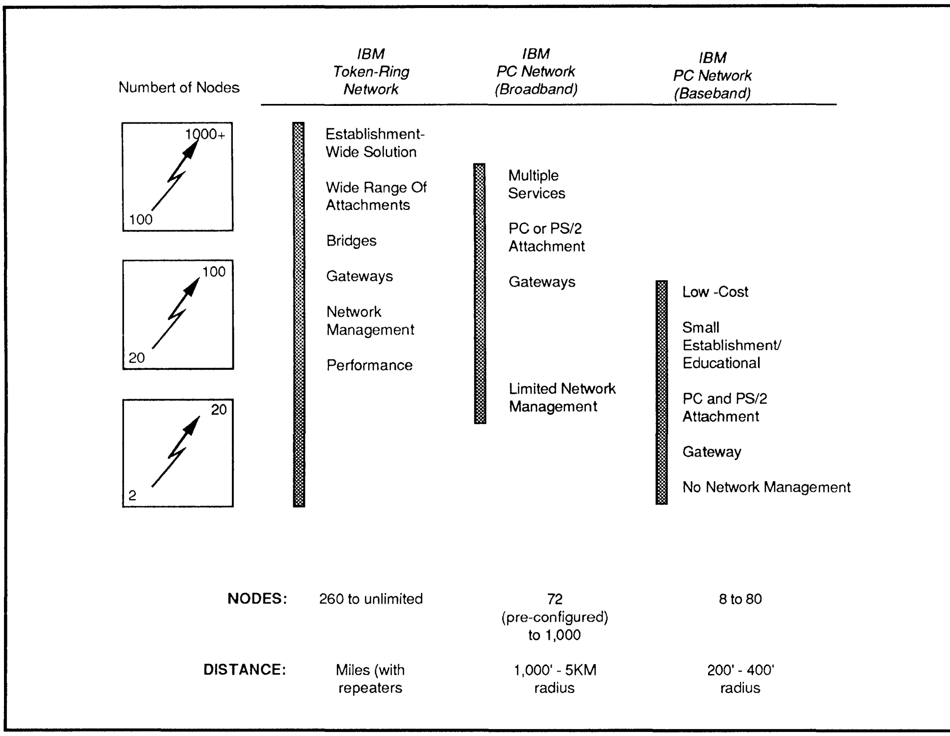

Figure 1-6: IBM LAN Positioning

SAA provides the foundation for IBM to enhance the consistency of IBM software products by providing a common programming interface, common communications support, and a common user access, and by offering common applications.

IBM Systems Application Architecture consists of four related elements. Two are completely new: Common User Access and Common Programming Interface. The third is based on extensions to existing communications architectures (Common Communications Support). These three elements establish the basis for the fourth, Common Applications, developed by IBM to be consistent across systems. Independent software vendors and customers developing applications for IBM's major systems will be encouraged to also use IBM SAA products.

Figure 1-7 contains a block diagram view of SAA. At the core of SAA are the communication products. Supported are X.25, SDLC, and the Token-Ring.

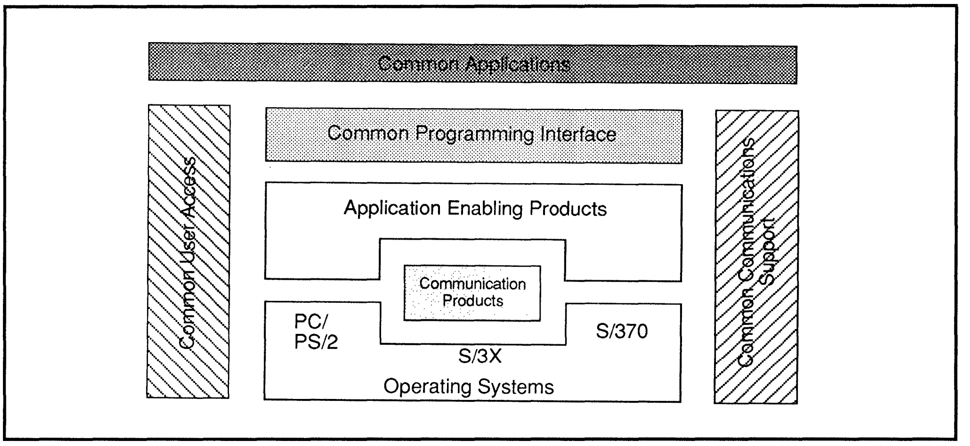

Figure 1-7: Systems Application Architecture

The SNA extended architectures of SAA extends peer connectivity to System/370 machines, PBX systems such as the IBM/Siemens CBX, departmental systems, and small systems network such as the low-entry networking (LEN) architecture, as well as support for the Token-Ring. The trend is toward smaller systems that are integrated together, and thus the network has to be more dynamic to accommodate the many small systems into the large logical whole. This extended SNA architecture will support dynamic connections, topology information, resource directory services, route selection services, and transparent management recovery.

The key to building logical communication services is Advanced Program-to-Program Communication (APPC). APPC is actually the marketing term for LU6.2. LU6.2 provides common interprogram communication services, common logical definition for peer connectivity and hierarchical connectivity, and data stream/device independence. Access to LU6.2 is typically provided by an Application Program Interface (API). LU6.2 is crucial in heterogenous IBM computing environments, since it provides the "glue" between the various computers for developing distributed applications.

In May of 1989, IBM announced its first major SAA family of applications called OfficeVision. The IBM OfficeVision Family of office systems applications consists of:

- OfficeVision/2 LAN Series
- OfficeVision/400 Series

- OfficeVision/VM Series
- Office Vision/MVS Series
- OfficeVision/VM-Asian Language Series
- OfficeVision/MVS-Asian Language Series

The IBM OfficeVision Family is the first offering in a new generation of Systems Application Architecture (SAA) application software designed to support office requirements. The family is an evolutionary step from existing IBM office products and provides integrated office applications, with emphasis on compatible function and data sharing.

Each series consists of a base product and separate optional products. The base provides connectivity and services for Operating System/2 (OS/2) Extended Edition and IBM Disk Operating System (DOS) workstations and enhances the capabilities for non-programmable terminals in the host series.

The OfficeVision/2 LAN Series introduces IBM office systems for the LAN environment. The function of the OfficeVision/2 LAN Series is also included in the OS/2 Office Feature and the OS/2 Office DOS Requester Feature of each host OfficeVision series.

User interaction is improved by exploiting the capabilities of both OS/2 Extended Edition and the Common User Access (CUA) graphical model with workplace extension delivered in the OS/2 Office Feature in each environment. IBM-developed applications, customer applications, and software vendor applications can be integrated using programming interfaces. Non-programmable terminals and directly connected IBM DOS and OS/2 Extended Edition workstations provide access to host office applications in each host series.

Highlights of the IBM OfficeVision Family include:

- SAA application providing an integrated family of office systems
- New CUA graphical model with workplace extension of OS/2 Extended Edition
- Support for OS/2 Extended Edition, IBM DOS, and non-programmable terminals
- Evolution from existing office products, with upgrade provisions for installed office system customers
- Broad range of office and decision support capabilities

- Architecture designed for continued growth and evolution
- Programming interfaces for customer business application integration
- National Language Support

The IBM OfficeVision/2 LAN Series is the OS/2 member of the IBM OfficeVision Family of office applications. IBM OfficeVision/2 introduces a new generation of SAA application software, offering a comprehensive set of office functions for OS/2 Extended Edition and IBM Disk Operating System (DOS) workstations. IBM OfficeVision/2 provides a Common User Access (CUA) graphical model with workplace extension implemented on OS/2 Extended Edition for use in each of the SAA environments of MVS, VM, OS/400, and OS/2.

The IBM OfficeVision/2 LAN Series provides office functions to interconnected OS/2 Extended Edition and IBM DOS workstations on a LAN. Support is introduced in IBM OfficeVision/2 Release 1 for mail, correspondence processing, address book, and file system. In addition to enhancing portions of IBM OfficeVision/2 Release 1, IBM OfficeVision/2 Release 2 adds the functions of calendar, decision support, file cabinet, library, extensive online help and tutorials, and an application platform. For the OS/2 user, IBM OfficeVision/2 establishes a new level of user interaction through a CUA graphical model with workplace extension and the general exploitation of OS/2 Extended Edition capabilities. IBM DOS users are presented with a user interface similar to that of the OS/2 user.

IBM OfficeVision/2 ties together other host- and OS/2-based applications. IBM OfficeVision/2 can distribute to and receive mail from VM, MVS, OS/400, VSE, and IBM LAN systems. In addition, the application platform assists in the integration of in-house and vendor OS/2-based applications.

The remainder of this report examines the key SAA transport mechanism -- the Token-Ring in great detail -- from adapter hardware to interfaces to applications to administration.

Chapter 2

Hardware

IBM Cabling System

The IBM cabling system has been designed to provide a structured wiring scheme that will work with all IBM communicating devices including Token-Ring networks. It is the intent of the cabling system to replace the ad-hoc wiring of systems which has occurred in the past with a new, moderate-cost, flexible wiring scheme. The system's components consist of: bulk cable, connectors, wall faceplates, wiring closet distribution panels, and cable assemblies (patch cables). The physical topology created by wiring for the Token-Ring resembles interconnected stars, in which all devices share the same dual twisted-pair bus that, if followed from device to device, forms a ring that eventually wraps back upon itself.

A device is connected to the cabling system as follows: A cable runs between the office (or work area) and a central wiring closet. The cable end in the office has a data connector attached to it that is mounted into a faceplate (either wall mount or surface mount). The cable in the wiring closet also has a data connector attached and is mounted on a distribution panel. The distribution panel, in turn, is mounted in a distribution rack above the MAUs. Patch cables link the data connectors on the distribution panel to the MAUs, creating the physical link between the cables. In the case of a small office network--fewer than eight stations--the wires can be connected directly to a single MAU which may be wall mounted in an optional housing.

The Cabling System supports existing non-token products through use of baluns (balanced-unbalanced cable impedance matching devices). The Cabling System can support existing point-to-point 3270 coaxial-based devices, AS/400, System/36 and Series/1 twinaxial-based devices, 5080 graphics systems, and loop systems, such as the finance communication system, programmable store system, and 8100 communication loop (based on a different token protocol) applications. These connections have nothing to do with the actual token-ring and still function as independent devices. The idea is to migrate devices such as the 3278-type terminals to a token-attached cluster controller such as the 3174.

Cable Types

For Token-Ring products, baluns are not needed. Devices attach directly to one of six copper cable Types -- Type 1, 2, 3, 6, 8, or 9. Type 5 cable, fiber-optic, can only be used with the fiber-optic repeater.

Type 1 is an overall, shielded data-grade cable with two solid twisted-pair 22 AWG wires. Type 1 is available as an indoor version with a braided shield or as an outdoor version with a corrugated metallic shield that is suitable for aerial installation or underground conduit. Type 1 indoor is also available in non-plenum or plenum versions.

Type 2 is basically a Type 1 indoor cable with four solid twisted-pairs of telephone grade (26 AWG) wire added around the outside of the shield. Type 2 is not available in an outdoor version. Type 1 cable is used exclusively for outdoor use.

Type 3 is basically (unshielded) telephone twisted-pair. Wire can be used where existing, un-used phone wire is already in place. A special jumper cable, consisting of Type 6 wire, a filter, and a data connector, must be used to connect Type 3 wire to a MAU. Type 66 connection block (also called a "punch-down block") is used to connect the Type 3 wire to the jumper. The jumper filter removes high-frequency components in order to meet FCC requirements. Figure 2-1 illustrates the various components in Type 3 wiring systems.

Because Type 3 is comprised of a small diameter, unshielded copper wire, it is subjected to many restrictions when used with the 4-Mbps Token-Ring network. The distance and number of devices it can serve is about a third of the limits for data grade (all other cabling system cable) wire. Type 1 and 2 cable is also guaranteed by IBM to support bit rates up to 16-Mbps -- Type 3 is not. Type 3 cable cannot be mixed with data grade wire (except when connecting two wiring closets in which Type 1 wire is used). When using Type 3

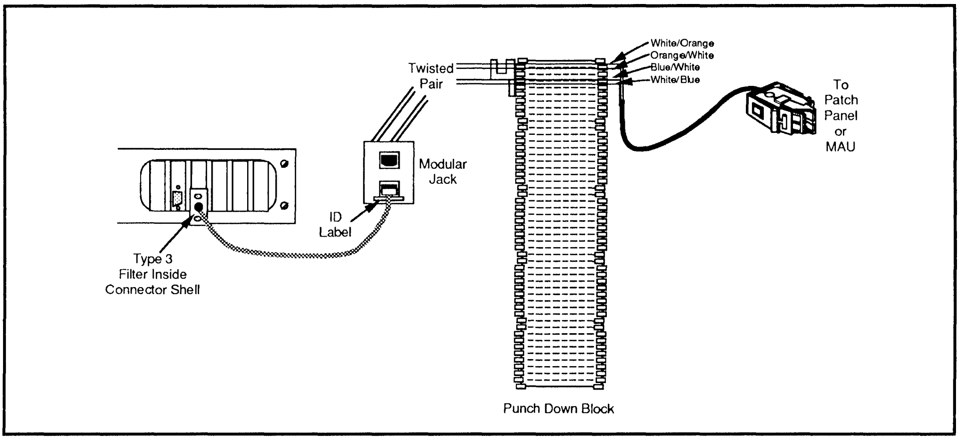

Figure 2-1: Type 3 Wiring Components

wire, care must be taken avoid intercoms, fluorescent lighting, power cables, arc welding equipment, heating equipment, electric motors, or any high voltage equipment.

For more details regarding the use of Type 3, refer to the IEEE draft, entitled "802.5B Recommended Practice for Use of Unshielded Twisted-pair Cable (UTP) for Token-Ring Data at 4 Mbits/sec".

Up to 260 devices can be attached to a cabling system configured for the Token-Ring using "data grade" (excluding Type 3 cable) cabling. If Type 3 cable is used, then the limit is 72 devices. Limitations are not dependent on the token protocol or capacity (loading) of the system, instead they result from an electrical phenomenon known as "jitter." The amount of jitter depends on the number of devices passed through by the data signal in the ring, as well as the length and quality of the wire.

IBM's support of Type 3 may not have been intended solely as an effort to counter the lower-cost AT&T PDS as mentioned in Chapter 1. Apparently, many IBM customers already had miles of unused Type 3 wire in place. The Token-Ring Type 3 support is designed to be ad hoc and temporary, with a migration path to the data grade wire that will be required by future 16-Mbps Token-Ring. If Type 3 is not already installed, IBM is discouraging its installation in favor of Types 1, 2, 5, 9, and so on. In fact, two categories of token-rings will be available -- Type 3 rings or Cabling System rings -- because Type 3 cannot be mixed with other cabling types except when connecting wiring closets together.

Type 5 is a 100/140 (100 micron core, 140 micro diameter) fiber cable that can be used with a pair of 8219 fiber-optic repeaters. The repeaters essentially bridge together data grade twisted-pair to fiber.

Type 6 is a data grade wire of stranded 26 AWG used for short runs in "patch" cables. Type 6 is often used to connect a device to a face plate which in turn connects to a Type 1 or 2 cable. In a small system, Type 6 can also be used to directly attach a device to a multistation access unit (discussed in Section 2.2).

Patch cables are made with Type 6 cable, and are available in lengths of 8, 30, 75, or 150 feet. The eight-foot version is used to connect a device adapter to a faceplate.

Type 8 wire is a 26 AWG parallel-pair data grade wire with a plastic "ramp" used to make under-carpet installation as unobtrusive as possible. Though Type 8 can be used similarly to Type 1, it can only service half the maximum Type 1 distance.

Type 9 wire is a 26 AWG shielded twisted-pair in a plenum jacket. It is designed to be used in environments where stringent fire codes are to be met, such as in plenums. Type 9 is essentially a lower cost version of Type 1 Plenum with a rating of two-thirds the distance capability of Type 1.

Figure 2-2 illustrates the different cable types.

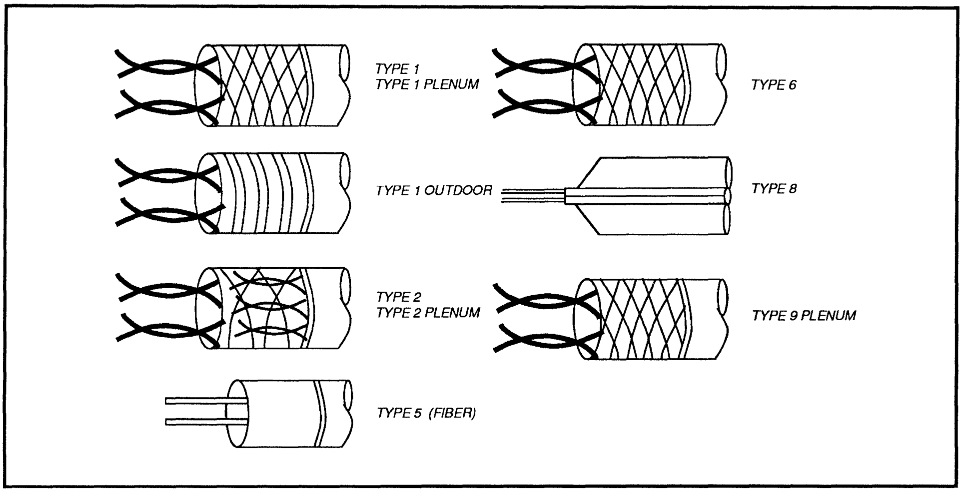

Figure 2-2: Cable Types

The table in Figure 2-3 summarizes the capability variations among the data grade Types 1 and 2 and the voice-grade Type 3 wire. The drive distances are relative to Type 1. The maximum drive distance (recommended by IBM) for Type 1 is 300 meters from a wiring closet to a work area and 200 meters between wiring closets. These are absolute maximum distances that become shorter as the physical ring wiring becomes larger (as more and more wiring closets, MAUs, lobe wiring, and devices are added). The IBM Token-Ring Network Introduction and Planning Guide provides tables and formulas that can be used to calculate the maximum distances.

	Cable Type						
	1	2	3	5	6	8	9
*Drive Distance **	1.0	1.0	.45	3.0	.75	.50	.66
Max. Data Rate (Mbps)	16	16	4	250	16	16	16
Devices Per Ring	260	260	72	260	260	260	260
Closets Per Ring	12	12	2	12	12	12	12
Voice Support	No	Yes	Yes	No	No	No	No

* Relative to Type 1

Figure 2-3: Performance Capabilities of Cable Types

Data Connector

The data connector is the plug which terminates all twisted-pair wire. Two data connectors can mate together by a 180-degree rotation of one connector -- thus, the same connector can be used throughout a non-Type 3 cabling system. When two connectors are used, one of the connections is contained in a faceplate. For Type 3 systems, faceplates containing RJ-11 jacks are available.

Copper Repeater

The IBM 8218 copper repeater extends the allowable distance between MAUs up to 750 meters (2,500 feet). Operating in pairs, 8218s redrive electrical signals on both the main ring path and backup path. Two crossover patch cables are required for each pair of repeaters. An individual 8218 amplifies and reclocks data transmission signals along the Token-Ring.

Typically these units are installed in a standard 19" rack (not supplied by IBM) or attached to flat surfaces such as walls or shelves. Rack installations require a rack mounting assembly accessory which accommodates up to seven individual IBM 8218s. The 8218s may be installed in or removed from a rack mounting assembly without powering down other 8218s in the same assembly. Individual 8218s snap into and out of the bracket.

The 8218s can operate in a Token-Ring environment which has Type 3 media installed. The 8218s are not supported for use on lobe wiring (the wiring from the MAU) to the Token-Ring-attached device. In this environment, two data grade media-to-type 3 filters are required where 8218s are used between 8228s, and Type 3 specified media is installed from the office to the wiring closet.

Fiber Repeater

The IBM 8219 Optical Repeater extends the allowable distance between MAUs up to 2,000 meters (6,600 feet). 8219 Optical Fiber Repeaters operate in pairs to convert data signals from electrical to optical and back.

The optical fiber repeaters are used in pairs. By operating in pairs, one 8219 converts electrical data transmission signals to optical signals in the main ring path of the network and the second converts the optical signals back to electrical signals. Individual 8219s can convert and redrive data transmission signals on the main ring path or the backup path. Like the copper repeaters, the 8219s are for use on lobe wiring.

Like the copper repeater, the 8219s can operate in an environment which has Type 3 specified media installed. A data grade media-to-type 3 filter is required for each repeater installed in this configuration.

The 8219s can also operate in IBM Token-Ring Network environments that have fiber cable other than IBM Cabling System Type 5 cable installed between 8228s. Fiber sizes that may be used (with restrictions) include 62.5/125 (AT&T), 50/125, and 85/125 micron. These restrictions are called out in the publication entitled IBM Token-Ring Network Optical Fiber Cable Options.

A pair of IBM 8219s requires:

- Crossover patch cable (one per fiber link)
- Two optical fiber BNC-to-biconic patch cables
- Two optical fiber dual socket clips
- Rack mounting assembly or surface mounting brackets
- Optical fiber biconic-to-biconic patch cable (2.5 m or 9.5 m) (optional)
- Data grade media-to-type 3 filter (Type 3 specified media environment at office wall)

Fiber Converter

The IBM 8220 Optical Fiber Converter, an electrical-to-optical and optical-to-electrical converter, operates at either 16 or 4-Mbps. It is an "intelligent" repeater, participating in polling and beaconing like other LAN stations on the ring. Two 8220s, connected by a fiber link up to 1.24 miles long, form a subsystem. The 8220 monitors the main token-ring and automatically wraps to the backup ring (a feature missing in the fiber repeater) if it diagnoses a fault, allowing the ring to remain operational for as many users as possible. If the 8220 is defined as a critical resource, the IBM LAN Manager (Version 2 or Entry) can monitor the 8220, detect any errors, and begin the problem determination process.

Multistation Access Unit

The IBM 8228 Multistation Access Unit (MAU) is a passive wiring concentrator that connects up to eight stations to 4- or 16-Mbps rings via drop cables (also called lobes). The design and operation of the MAU is what gives the network's physical topology its "star-wired ring" name. A ring in jack on the left-hand side of the device and a ring out jack on the right-hand side of the device provide for a daisy chain connection to other MAUs. The MAU provides for insertion/bypass of lobe segments and associated attaching devices. In addition, it also facilitates wire fault detection by an attached device, such as the IBM PC Token-Ring Adapter.

The MAU can be used in many different physical configurations of the ring. Figure 2-4 shows a network with a single MAU and one with multiple MAUs. A device attaches to the MAU using a Type 6 cable. If the MAU is in a wiring closet as illustrated in Figure 2-5 then a patch panel is needed to jumper from Type 1 or 2 cable to Type 6. If Type 3 wire is used, a special filter jumper is required.

The attached device (i.e., token-ring adapter), is responsible for activating a relay within the MAU to switch itself into the ring. This is done by maintaining a "phantom" voltage component on the lobe wire. This voltage charges a capacitor on the relay which then inserts the lobe into the network. When the device fails or is turned-off, the capacitor will de-energize and latch out the device. A look at the operation inside the MAU is shown in Figure 2-6. Note the automatic wrapping feature of unconnected jacks.

The MAU can be rack or wall mounted, as illustrated in Figure 2-7. The wall mount feature requires an optional wall mount housing, available from IBM. For very small networks (eight or less nodes) a single wall mount MAU can be used with a maximum of 150 meters of Type 6 cable going directly to the device or wall plate.

Though the relays automatically latch when devices are inserted/removed, it is possible during shipment of the MAU, for a sudden

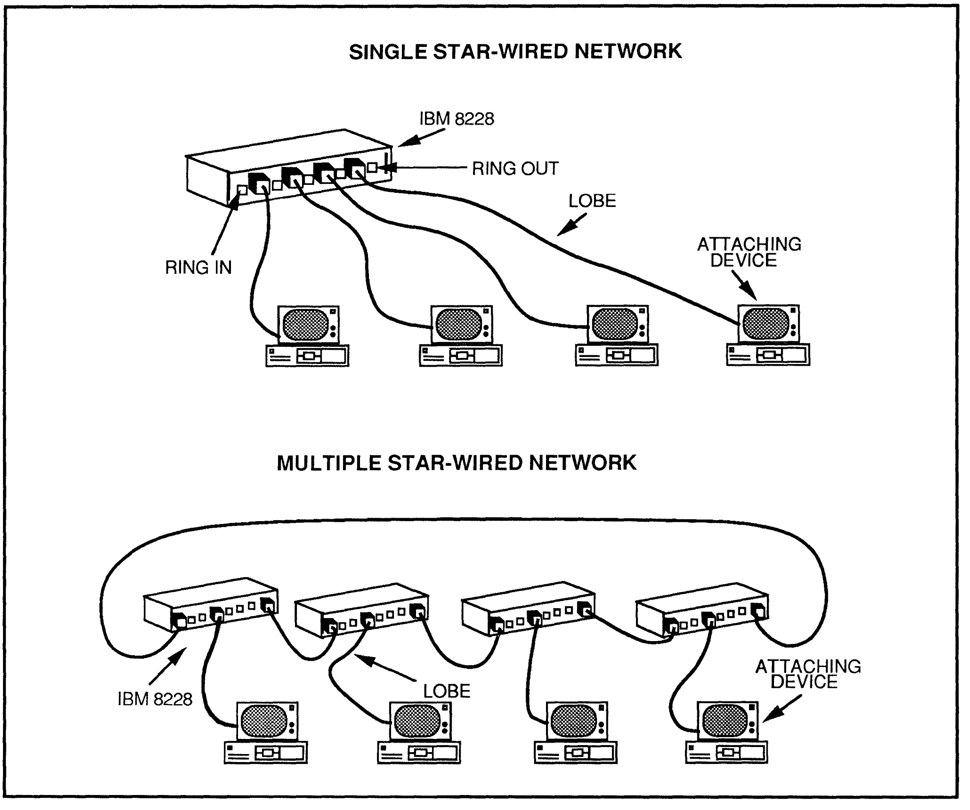

Figure 2-4: MAU Configurations

movement (such as dropping it) to cause a relay to be latched in the wrong state. IBM provides a testing tool with the MAU. Insertion of the tool into the connected ring tests whether all relays contained are wrapping. If an anomaly is detected, the tool can be inserted into each lobe connection to test the associated relay's status.

There are numerous third-party alternatives to IBM's 8-port MAU. An interesting variation is a "daisy-chaining" MAU such as the 3Com Ring Tap product. The Ring Tap is essentially a single-port MAU, such that one could configure the physical topology as a ring as opposed to a star-shaped ring. Although wiring is simplified, it becomes hard to manage (no centralized isolation points) and adding new Taps can be disruptive.

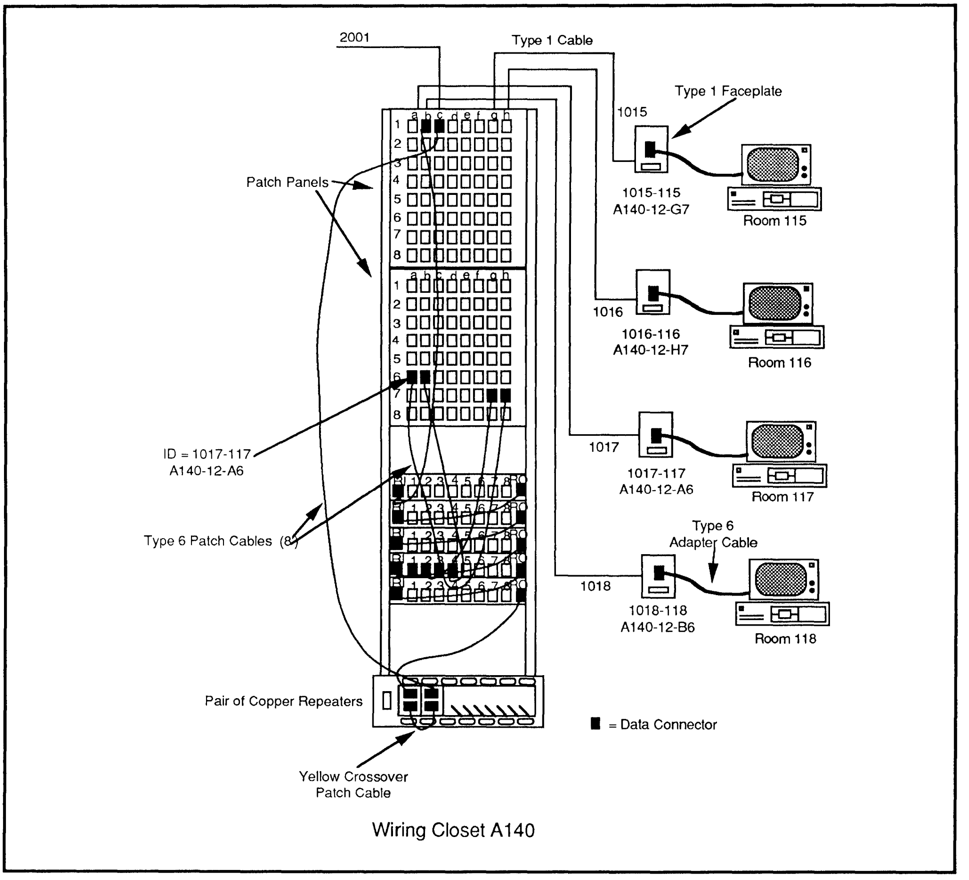

Figure 2-5: MAU Rack Configuration with Repeater

At the other extreme, there are "intelligent" MAU's such as the Ungermann-Bass Distributed Wiring Concentrator (DWC). The DWC offers ten ports (excluding ring in and ring out) and can operate in an active or passive mode (IBM MAU compatible). In the active state (power applied), it allows the U-B Network Management Console (NMC) to perform problem determination, fault isolation and when appropriate, remove stations from the ring.

A summary of the evolution in the IBM Cabling System is illustrated in Figure 2-8.

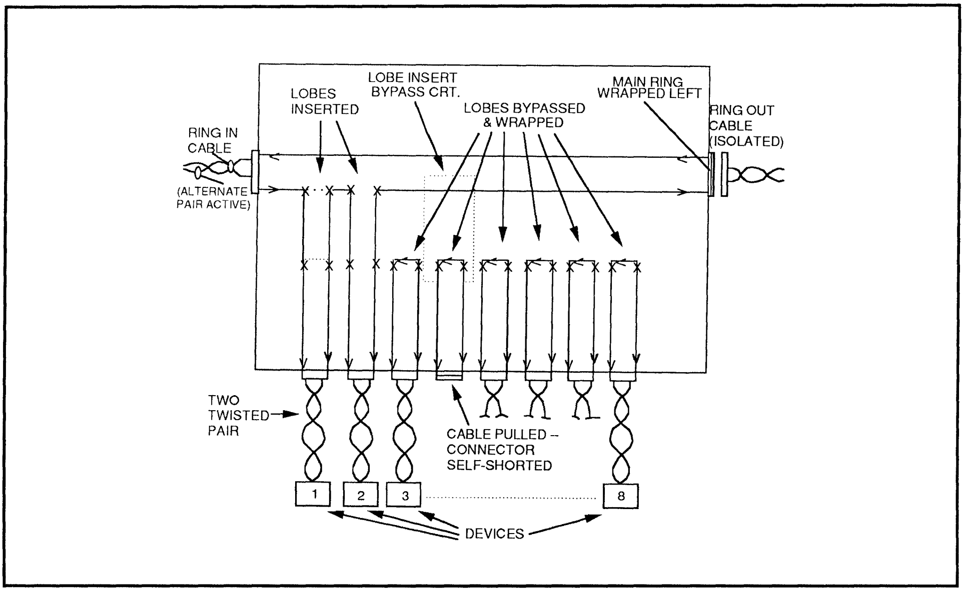

Figure 2-6: Inside the MAU

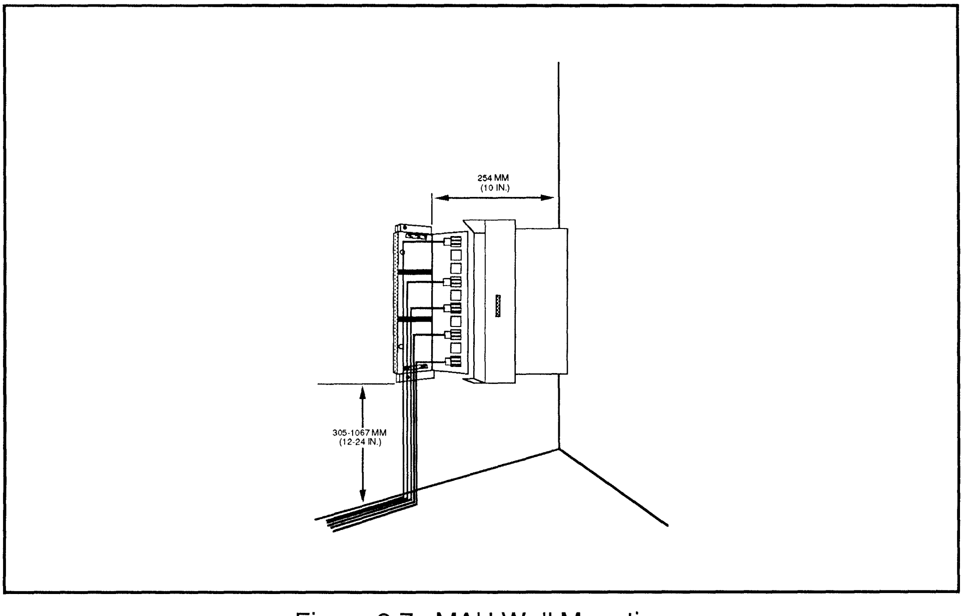

Figure 2-7: MAU Wall Mounting

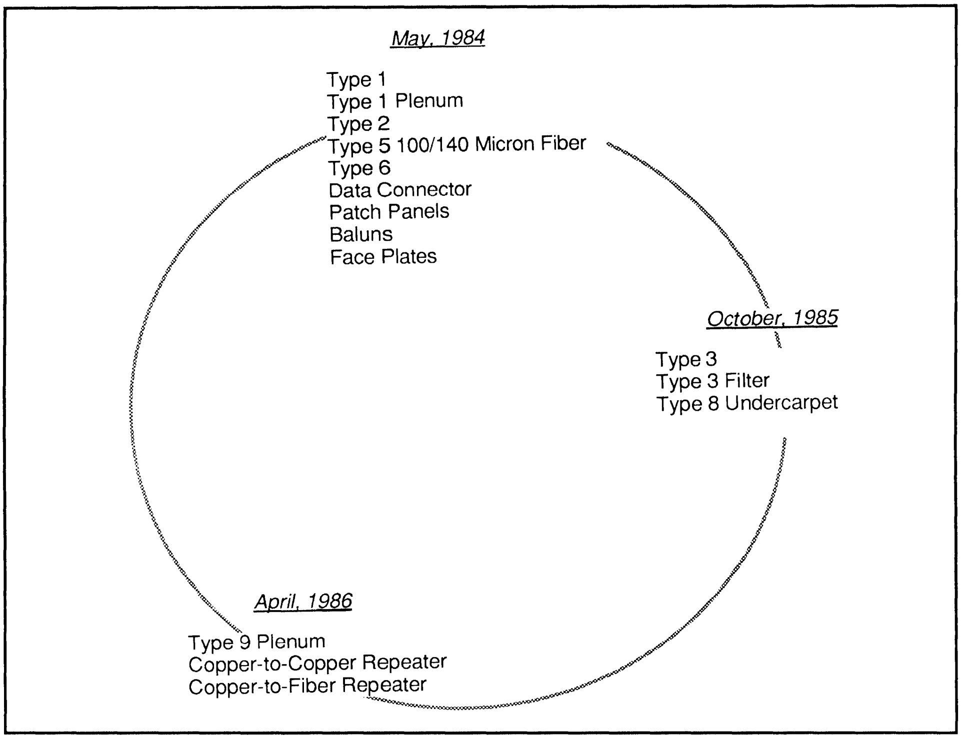

Figure 2-8: IBM Cabling System Evolution

16-Mbps Over Unshielded Twisted-Pair

There are several vendors offering MAU support for 16-Mbps over unshielded twisted pair, IBM not being one of them!

One such vendor is Proteon who is offering intelligent wire centers that support unshielded twisted-pair (UTP) cabling over 16-Mbps (and 4-Mbps) token-ring networks. Proteon's products feature unified media support and network management capabilities.

The key to Proteon's UTP support is a multi-access unit (MAU) called the Series 70 Intelligent Wire Center. This modular hub presently supports 4-, 10- (for the proNET 10 token-ring), and 16-Mbps networks on UTP and shielded twisted-pair (STP) cables, as well as fiber optics.

The Series 70 is compatible with the AT&T Premises Distribution System (PDS) and the IBM Cabling system. Proteon claims that PDS is often considered to be the best way to run networking cable when the wiring system is in place for telephone communications.

Proteon recommends passive filtering as a simple and pragmatic way to address the technical challenge of 16-Mbps running over UTP: passive filtering at both ends of the UTP cable eliminates electro-magnetic interference (EMI) and facilitates reliable signal transmission, says Proteon. Proteon's filtering approach also limits radio frequency interference (RFI) within the acceptable frequency range.

Proteon's 16-Mbps UTP networks will support distances of at least 85 meters (about 279 feet) between the wiring closet and the workstation. According to AT&T surveys, this distance covers 99% of practical distances between network wire closets and workstations. Proteon supports at least 72 devices on 16-Mbps UTP networks.

The Series 70 uses an independent, out-of-band communications channel for management command and control. This channel enhances the network management capabilities already built into the IEEE 802.5 standard. Network managers using the Series 70 with Proteon's TokenVIEW-4 Network Management software can reconfigure the network to partition large portions of the LAN or individual PCs through the out-of-band channel. Faulty network components and broken cables can be diagnosed and isolated from the rest of the network in five minutes or less, even when the network is down.

Proteon is one of several vendors trying to iron out a workable standard for 802.5. Together with AT&T, Proteon has proposed to the IEEE 802.5 standards committee that they define the technical specifications for incorporating 16-Mbps UTP into the 802.5 standard.

Of Adapters and Chips

4-Mbps token-ring adapters are available for the following machines:

- 3174*
- 3720
- 3725
- 3745*
- 4702
- 8100
- 9370*
- AS/400

- Eclipse
- EISA*
- ISA (PC, AT)*
- Macintosh
- Micro Channel (PS/2, some 9370's, RISC System/6000)*
- Multibus
- Q-bus
- RISC/6000
- RT/PC
- Series/1*
- System/36 (5363)
- VMEbus

Those marked with an * are also available in 16-Mbps versions. For a complete list of adapter offerings from various vendors, consult Appendix C. Here, we will examine the IBM PC and Micro Channel adapters in more detail.

The First Token-Ring Adapter

The original Token-Ring adapter card for the IBM Personal Computer is a full-sized (4.2 inches by 13.2 inches) card (see Figure 2-9) that works with industry standard architecture (ISA) buses (i.e. the PC and PC/AT). Although no longer available, this adapter was the first to use IBM's 4-Mbps token-ring chip set upon which other all currently available IBM 4-Mbps adapters are based.

Five custom analog and VLSI devices handle the protocols and interface to the two twisted-pair. The non-analog devices were developed by IBM Burlington, using high-density, high-speed bipolar technology. An on-board 16-bit processor aids in initial diagnostics, on-going diagnostics, and communications functions.

The microprocessor executes resident (on-board) microcode (32K 16-bit words arranged as two 32K by 8-bit EPROMS) which provides the host access to the data link functions per IEEE 802.2 LLC or physical link functions per IEEE 802.5.

When an IBM token-ring adapter is turned on for the first time, diagnostics built into the Token-Ring adapter perform a power-on-self-test (POST) procedure. This will check all the internal operations of the adapter, including the on-board timers. The adapter also checks

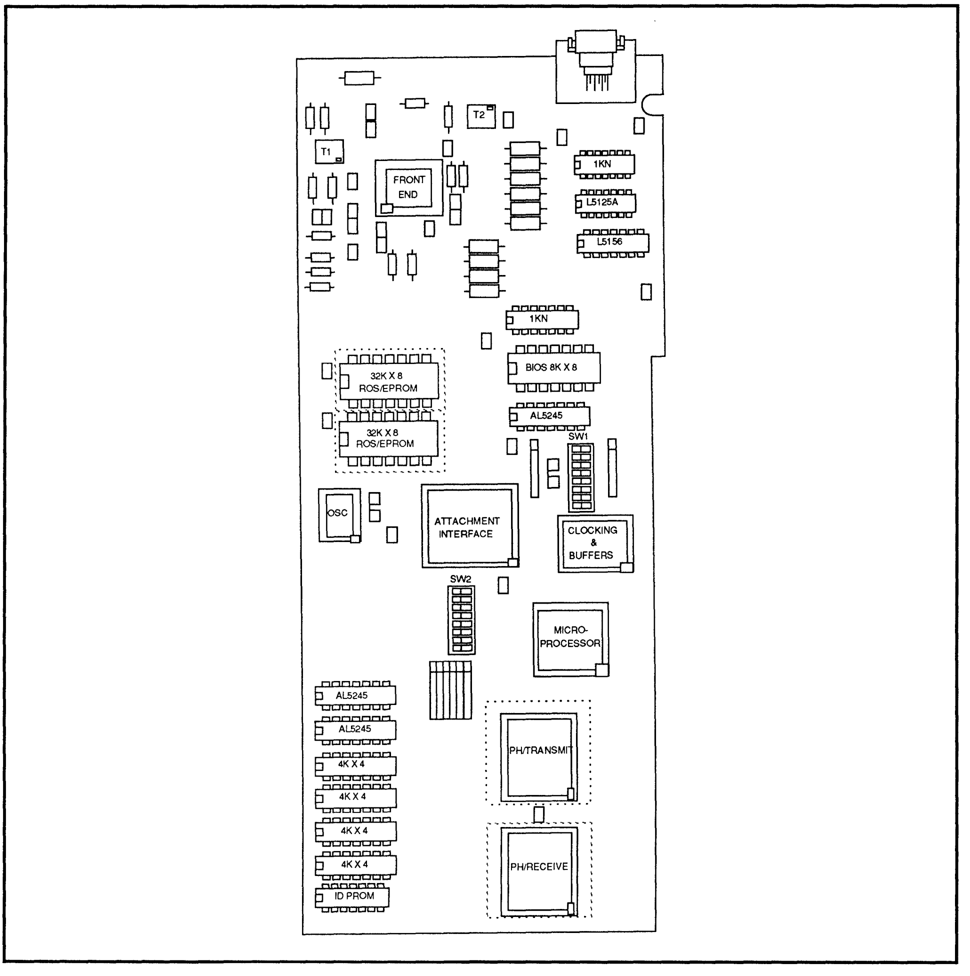

Figure 2-9: Token-Ring PC Adapter Card

the lobe cabling (up to the MAU and back) with loop-back tests to ensure that the cable is performing properly.

A single chip functions as a front-end. It is essentially an analog device which performs differential Manchester encoding/decoding, data synchronization, and physical insertion/removal from the ring. The chip is transformer-coupled, to electrically isolate the adapter from the cable.

The two-chip (one for transmit, one for receive) protocol handler performs parallel-to-serial conversion, encoding/decoding of data (from the front-end), cyclic redundancy check (CRC) generation and checking (removal), DMA to shared memory, monitor function, and detection of ring errors.

The shared memory is organized as four banks of 4K by 4-bit static RAM. It can be accessed as 8K by 8 from the host, or 4K by 16 from the adapter's microprocessor. The shared memory starting address is programmable by the host, eliminating the need to set switches. Any 8K boundary can be programmed, and certain areas of the shared memory can be protected (set to read-only) from corruption by the host. Figure 2-10 illustrates how the shared memory can be mapped into the PC memory domain, and how one might protect certain areas.

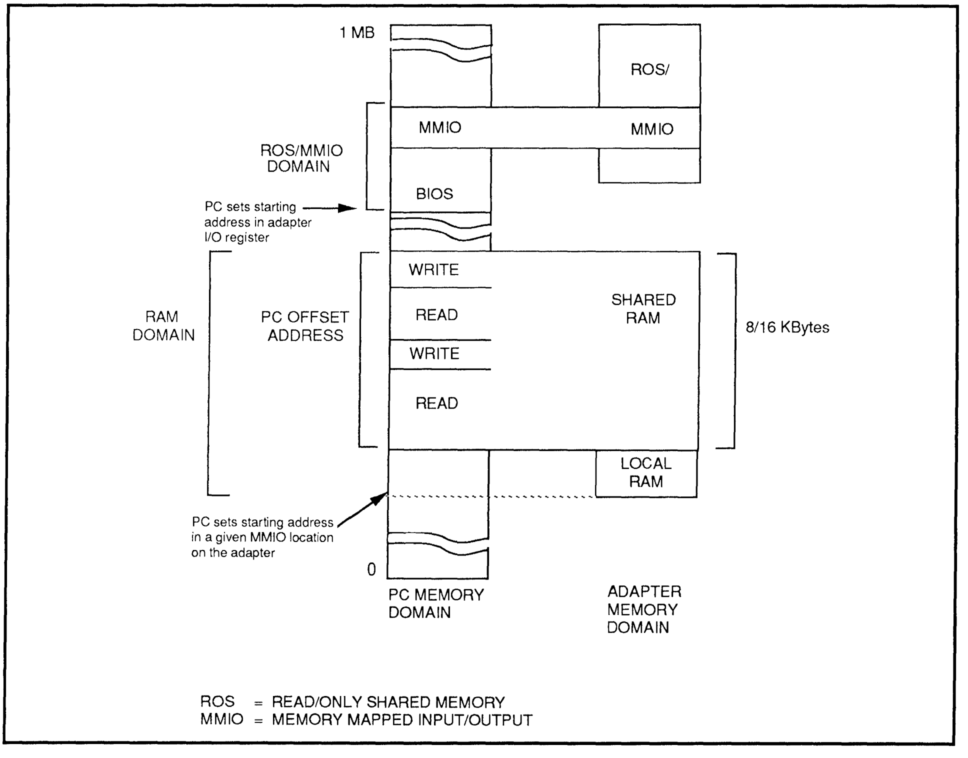

Figure 2-10: Token-Ring PC Adapter Memory Structure

With the introduction of the PC Token-Ring adapter II, 8 Kbytes of static RAM was added. In addition, there were changes made to the microcode to support the IBM Attachment/36 and Token-Ring Bridge products.

In addition to the shared memory interface (also referred to as the memory mapped input/output (MMIO) interface), certain functions are controlled by the programmable I/O (PIO) interface via an I/O location in the PC's I/O space. The address of this I/O location is set via a dip switch on the adapter. This I/O port gives access to one of sixteen 8-bit control registers. Functions provided by the control registers include: bi-directional interrupt and status; the PC shared RAM starting address; the PC shared RAM management (i.e., setting protected areas) timer control which provides millisecond (ms) level timing; and the PC timer value register.

Five timers are provided by the adapter to supply interval and dead-man timings. The interval timers ensure proper token operation of the ring, as well as a general-purpose 10 ms timer accessible by the host PC. The dead man timer is a 120 ms timer which checks if the adapter code is executing. If the timer expires, a procedure is initiated to get the bad adapter physically off the ring.

To connect to the network, the adapter card provides a widely available DB-9-type connection to the PC adapter cable. Pins 1 and 6, 5 and 9 are used to connect to the transmit +, transmit -, receive +, and receive -. The adapter cable consists of a flexible 8-foot, Type 6 cable for connecting to data grade (Type 1 or 2) media. An optional Type 3 media filter is available for use with Type 3 telephone-type wire. In addition, the adapter supplies the phantom voltage required to drive the insertion relay in the MAU.

PS/2 Adapter

The IBM Token-Ring Adapter/A is used for the Micro Channel found in Personal System/2 Model 50 and higher. It contains the same basic chip set as described above and has 16 Kbytes of static RAM. About the only real enhancement to this adapter is that it is a 16-bit interface to the host and has a provision for remote program load (RPL), a feature for diskless or network boot (where the operating system is downloaded from a file server) operation.

16/4 Adapter

The 16/4 Token-Ring Network adapter cards feature the first analog/digital chip built with IBM's one-micron complementary metal oxide semiconductor (CMOS) technology. Taking advantage of the cooler-running CMOS technology, IBM engineers at Research Triangle Park, NC, and at Essex Junction, VT, jointly developed the new 40,000-circuit interface chip, integrating functions that previously required six chips. This increased integration enhances reliability and allows the new 4-Mbps adapter to be packaged in a half-size card.

The 16/4-Mbps adapter cards provide 64 Kbytes of RAM; as previously noted, IBM Token-Ring Network adapters used 8-Kbyte or 16-Kbyte RAMs. The 64-Kbyte RAM on the 16/4 adapters permit the use of larger data frames and allows more concurrent sessions to be established.

The IBM Token-Ring Network 16/4 Adapter and 16/4 Adapter/A, permit attachment of IBM PC's and compatibles (with ISA bus) and IBM PS/2 and compatibles (with Micro Channel bus) to the token-ring with switchable data rates of either 16 or 4-Mbps (the adapters will operate with older 4-Mbps adapters at the 4-Mbps rate). A remote program load is available as an option.

The Texas Instruments' Chip Sets

While both the IBM and TI chip sets provide a functional 802.2 and 802.5 interface, there are several subtle differences between the two implementations. TI offers both 4- and 16/4-Mbps chip sets.

TI and IBM collaborated during the development of the TI chip set with IBM supplying (and changing several times!) the functional spec. The joint TI-IBM agreement was announced in September of 1982, and the TI chip set was formally announced on October 15, 1985, the same day as IBM's formal Token-Ring announcement. A system developed with TI's chip set is fully compatible with IBM-developed Token-Ring adapters. Over 200 vendors have requested TI chip set evaluation kits and over a dozen vendors, including 3Com, Madge, Proteon, and Racore, have developed adapters with it.

The 4-Mbps TI chip set is a five-chip implementation. The entire TI chip set is designated as the TMS380 LAN Adapter Chip set. The TMS38030 System Interface is analogous to the IBM Attachment Interface, the TMS38010 Communications Processor Communications to the IBM 16 bit processor, the TMS38020 Protocol Handler to the twin IBM protocol handler chips, and the TMS38051 Ring Interface transceiver and TMS38052 ring interface controller to the IBM front-end.

The IBM chip set utilizes a shared memory interface whereas TI depends on a direct memory (DMA) interface. The TMS380 reference book includes an example design for a PC AT adapter card. A pictorial summary of the TMS380 chip set is shown in Figure 2-11.

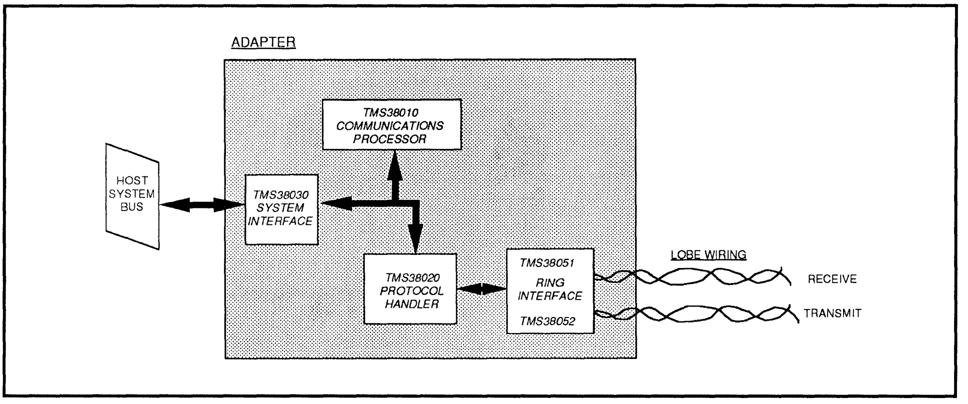

Figure 2-11: TI Token-Ring Chip Set

The TMS38010 Communications Processor (CP) is a dedicated 16-bit CPU with 2.75 Kbytes of on-board RAM. The CP executes the firmware contained within the 16 Kbyte ROM on the TMS38020. The CP can additionally access up to 44 Kbytes of external RAM/ROM. A ROM option is available that enables the TMS38010 to execute the IEEE 802.2 Logical Link Control (LLC) Type 1 (connectionless) and Type 2 (connection-oriented) protocol. The LLC code was rewritten by TI from source code supplied by Sytek.

The TMS38020 Protocol Handler (PH) executes the functions of IEEE 802.5, LAN management services, ring operation, and diagnostics of the chip set. Per the 802.5 specification, the PH implements the Manchester encoding/decoding and frame-address recognition (group, local, and functional). The PH contains four DMA channels, two for transmit and two for receive. All data paths and registers in the chip are parity-checked for increased reliability and integrity of its operation.

A special version of the TMS38020 called the TMS38021 enhances performance for bridge applications by fully supporting IBM source routing (source routing is covered in detail in Chapter 3).

The TMS38030 System Interface (SIF) provides up to 40 Mbps data transfer via its own built-in DMA controller. The host system bus interface can be programmed to handle two major families of processors; the Intel 8086 family and the Motorola 68000 family. The DMA feature can handle linked lists within the host memory. Commands are sent from the host to the SIF via command blocks with a high-level command structure. Example commands are TRANSMIT, RECEIVE, and READ ERROR LOG.

The ring interface is handled by the TMS38051 Ring Interface and TMS38052 Ring Interface Controller. The Ring Interface contains a phase lock loop for clock recovery, data detection, and phase alignment. It provides the clock for the ring when in active monitor mode, and the phantom drive signal to the MAU, a loop-back path for diagnostics, and error detection of wire faults.

Per the IBM specification, the TI chip set has the capability to allow the LAN to recover from soft (i.e., corrupted or lost tokens) and hard (i.e., a bad lobe wire or transceiver on the adapter card) errors with minimal impact. The error is automatically logged and the token operation is recovered. The error condition is then reported to the host for appropriate action.

To aid in reducing the package count when designing a token-ring adapter, TI offers the ASIC-LAN Toolkit (ASIC is an acronym for Application Specific Integrated Circuit). The kit contains captured schematics with simulation and test vectors in order to shorten the development time of token-ring ASICs. It provides 30 soft macro (defined in TI's SystemCell 2-micron standard cell library) building blocks and four example ASIC designs. TI estimates that two to three months can be saved in design time.

Announced at the same time as IBM's 16-Mbps adapters, TI's newer TMS380C16 COMMprocessor for 4 and 16-Mbps token-rings, performs functions which previously required five chips. It integrates the functions performed by three of the five members of the first-generation TMS380 token-ring chipset: the TMS38010 Communications Processor, the TMS38020/21 Protocol Handler, and the TMS38030 System Interface.

Additionally, two support circuits used with the first-generation chipset, both VLSI standard-cell (ASIC) components, have been incorporated in the token-ring COMMprocessor: the PC bus-interface unit (BIU) and the DRAM memory-expansion unit (MEU).

The other two members of the first-generation TMS380 chipset, the TMS38051 and TMS38052 Ring Interface, have been integrated on a new single-chip, IEEE 802.5-compatible ring interface.

Fabricated in one-micron EPIC CMOS technology, the token-ring COMMprocessor will be available in two versions. The TMS380C16 supports both the 16-Mbps ring data rate proposed by

IBM and the IEEE 802.5 subcommittee for higher network performance and the industry-standard 4-Mbps rate. A 4-Mbps-only design will be introduced at a later date.

The single-chip TMS380C16 COMMprocessor reduces to less than 10 square inches the space required to implement a full-function, IBM-compatible token-ring adapter for a PC/XT or compatible. This supports the design and production of compact, cost-effective PC motherboards, workstation motherboards, and 1/4-size PC cards. The COMMprocessor also dissipates less than one-fourth the power of the three NMOS and two CMOS devices it integrates: this makes the COMMprocessor chip suitable for use in low-power applications such as portable PCs and peripherals.

Many third parties, including LAN OEMs and computer manufacturers, have begun TMS380C16-based designs. Over 50 companies are working on or have announced token-ring products incorporating TI's token-ring COMMprocessor, in an effort to accelerate the market for 16-Mbps token-ring LANs. Some of these companies include: Codenoll Technology Corporation; FORMATION, Inc.; INTERLAN, Inc.; Local Data, Inc.; Madge Networks Ltd.; McDATA Corporation; NCR Corporation; Network General Corporation; Olicom A/S; Proteon, Inc.; Pulse Engineering; Racore Computer Products, Inc.; Silcom Manufacturing Technology, Inc.; Simpact Associates, Inc.; SynOptics Communications, Inc.; Sytek, Inc.; and Wang Laboratories.

The COMMprocessor performs all the functions of the combined first-generation devices and maintains IBM and IEEE 802.5 compatibility. The Communications Processor section of the TMS380C16 contains a dedicated 16-bit CPU that executes the adapter firmware supporting IEEE 802.5 MAC protocols and the IEEE 802.2 LLC protocols. The Protocol Handler section includes hardware-based IEEE 802.5 protocol state machines and the Token-Ring Network, including both 16 and 4-Mbps. The System Interface section is a high-speed, bus master, linked-list DMA interface that provides up to 10 Mbps of data transfer bursts to the host system.

COMMprocessor enhancements include increased CPU performance and a 2-Mbyte adapter local address space with a glueless interface support for 256K, one-megabit, and four-megabit DRAMs. The extended 32-bit system address gives LAN adapters access to the entire memory space of host systems based on high-performance 80386, 68030, and RISC processors. The high-speed, bus master, linked-list DMA provides up to 10 Mbps of data transfer bursts, which is suited to match such popular high-speed buses as IBM's Micro Channel, TI's Nubus, VMEbus, and the new EISA bus.

Also announced by TI in conjunction with the COMMprocessor is the TMS38053, a new single-chip, IEEE 802.5-compatible ring interface designed for use with the TMS380C16 COMMprocessor in 16-Mbps token-ring LAN adapters. Fabricated in Advanced Low-Power Schottky (ALS) technology, the TMS38053 integrates the functions performed by TI's TMS38051 and TMS38052 and other typical ring interface circuitry. It includes the digital and analog circuitry to connect separate transmit and receive channels, thus enabling simultaneous transmission and reception of frames. A single-chip ring-interface device that supports both 16- and 4-Mbps operation will follow.

The COMMprocessors are packaged in plastic, surface-mountable, 132-pin JEDEC quad flat packs (QFPs); the ring interface is packaged in a 44-lead plastic leaded chip carrier (PLCC). The TMS380C16 and TMS38053-16 are available in the 16-Mbps token-ring design kits. Volume production for these devices was targeted for the second quarter of 1989, but has been pushed a year to 1990.

The Toshiba Chip Set

Toshiba's TC35802G TRC (Token-Ring Controller) is an IEEE 802.5-based LAN controller designed to interface to the Intel iAPX series of microprocessors. The TRC's built-in DMA controller and linked list buffer management scheme allows a system to be configured for high performance in node processor applications. For systems without a dedicated node processor, the system designer can choose between shared memory interfaces and bus master applications.

All MAC-level 802.5-defined functions are supported. The Logical Link Control, Configuration Report Server, Ring Error Monitor, and Ring Parameter Server are implemented in software. The TB32042F MDR (Media Driver/Receiver) allows for easy connection between the ring network and the TRC. The MDR has built-in phantom drive control, loop-back, watchdog timer, and wire-error detection, and it is software switchable between 4-Mbps and 16-Mbps speeds.

Figure 2-12 shows an intelligent-type Token-Ring Adapter used to connect PC AT buses. In this example, the adapter is configured with the TRC (T5B79), media driver (TB32042), and iAPX186 network processor, a ROM containing software for the TRC controller, a buffer and work area SRAM, and an interface circuit for connecting PC AT buses.

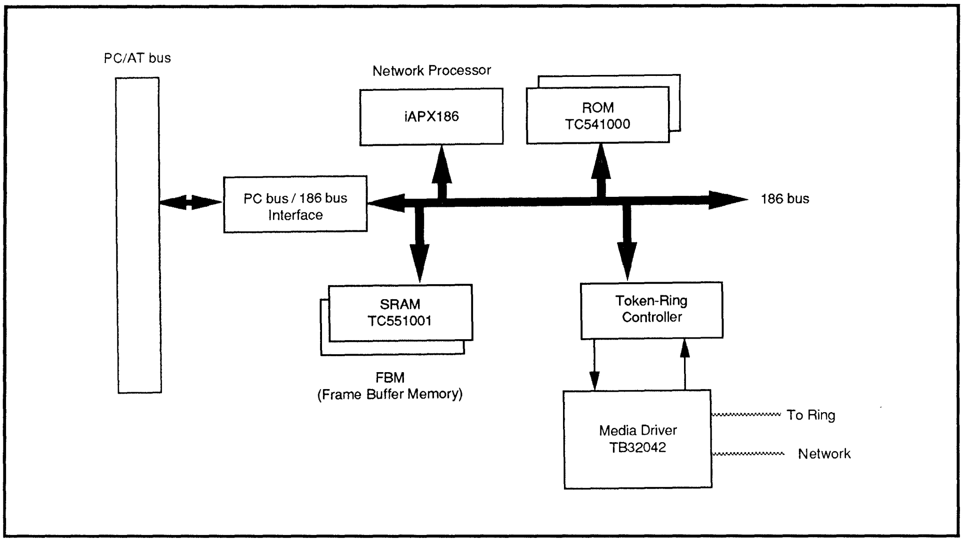

Figure 2-12: Toshiba Token-Ring

In addition to its ring station functions, the TRC also has bridge station functions for connecting to other rings.

Toshiba appears to be the first semiconductor company to capitalize on TI's 16/4-Mbps chipset delay. Expect other semiconductor companies, such as Western Digital, to soon provide token-ring chipsets. LAN vendor Ungermann-Bass has been using its own token-ring chips but operating at only 4-Mbps.

With multiple vendors entering the token-ring chipset fray, expect things to get more interesting in the token-ring marketplace in general -- commodity pricing leading to low-cost, 16-Mbps adapters, performance and architecture comparisons, ease of design, compatibility with existing IBM and third-party token-ring software, adapter interoperability, etc.

The Mainframe Connection

3720

The IBM 3720 Communication Controller and IBM 3721 Expansion Unit are a lower-cost, lower capacity entry in the 3725 communication controller family. A link-attached model has been specifically designed for remote operation.

The 3720 provides four host attachments via two channel adapters and two 2- processor switches, up to 60 lines, and a maximum of two

IBM Token Interface Controllers (TIC). Specifically, Models 11 and 12 support the Token-Ring. Advanced Communication Function/Network Control Program (ACF/NCP) Version 4 Release 2 or greater is required for the 3720. Alert support for the Token-Ring is via the hardware monitor component of NetView; NCCF/NPDA does not support alerts from the Token-Ring. NetView is discussed in greater detail in Chapter 7.

3725

Special hardware features support direct attachment of the 3725 front end processor (FEP) controller to the Token-Ring. The Token-Ring Subsystem permits attachment of appropriately equipped 3725s as well as 3726s to IBM Token-Rings. The subsystem consists of Line Attachment Base (LAB) Type C and Token Interface Controller (TIC).

The Line and Token-Ring Attachment Base provides a Token-Ring multiplexer and a physical base for up to four TICs. In addition, each LAB Type C provides a communication scanner and a physical base for up to 16 line attachments. The TIC provides one attachment to the Token-Ring. It contains a microprocessor operating under control of a resident microcode. Under control of ACF/NCP, the coupler provides logical link control functions conforming with the IEEE 802.2 standard.

One LAB Type C can be installed in a 3725 Model 1 or 2. In a 3725 Model 1 with a 3726, one LAB Type C can be installed in the 3726 if a LAB Type C is installed in the 3725, or two in the 3726 if no LAB Type C is installed in the 3725.

Figure 2-13 summarizes the key features of the 3725 and 3720.

3174

The IBM 3174 Subsystem Control Unit (cluster controller) large cluster Models 1L, 1R, 2R, and 3R attach up to 32 3270 Information Display System terminals. The small cluster Models 51R, 52R, and 53R attach up to 16 terminals. These control units provide attachment of 3270 system displays, printers, and workstations to IBM host processors via a local channel, remote link, Token-Ring, and IBM Token-Ring LAN gateway (see Figure 2-14). An optional feature permits attachment of ASCII terminals and attachment to ASCII hosts via telecommunications links.

The 3174 models are functionally equivalent to the IBM 3274 Control Unit Models 41A, 41C, 41D, and 61C but offer improved price/performance, usability, and increased functional capabilities.

	3725-1	3725-2	3720
Storage	.5 Mbtes to 3 Mbytes	.5 Mbtes to 3 Mbytes	1 Mbtes to 2 Mbytes
Maximum Duplex Line Attachment	256	80	60
Maximum Line Speed	256 Kbps	256 Kbps	256 Kbps
Host Attachments	8	4	4
Maintenance & Operator Subsystem	Yes	Yes	Yes
Direct Attach Maximum Feet	492	492	492
Token-Ring Adapter	8	4	2
Line Interface	EIA RS-232-C, 366A CCITT V.24, V.25, V.35, X.21 Wideband Direct Attach Very High Speed Adapter (RPQ)S	EIA RS-232-C, 366A CCITT V.24, V.25, V.35, X.21 Wideband Direct Attach	EIA RS-232-C, 366A CCITT V.24, V.25, V.35, X.21 Wideband Direct Attach

Figure 2-13: 33725 & 3720 Features

It is the 3174 Models 3R and 53R that provide direct host attachment via the Token-Ring. These models communicate with the host via an IBM 3720 or 3725 with the NCP/Token-Ring interconnection facility interconnection of ACF/NCP Version 4 Release 2, or via a 3174 Model 1L with the IBM Token-Ring Network 3270 Gateway Network optional feature.

As an option, IBM Token-Ring Network 3270 Gateway is available for a 3174 Subsystem Control Unit Model 1L using SNA protocols. This option provides the capability for up to 140 Token-Ring attached devices, as PU 2.0 nodes, to communicate with an IBM host. Any combination of the following Token-Ring attached devices are supported by this feature:

- 3174 Model 3R or 53R
- PC using IBM Personal Computer 3270 Emulation Program Version 3.0
- PC using APPC/PC (as a PU 2.0 node)
- System/36 with the LAN Attachment Feature and using 3270 emulation or APPC (as a PU 2.0 node)

The devices and workstations directly attached to the 3174 and those attached via the Token-Ring can coexist and operate concurrently.

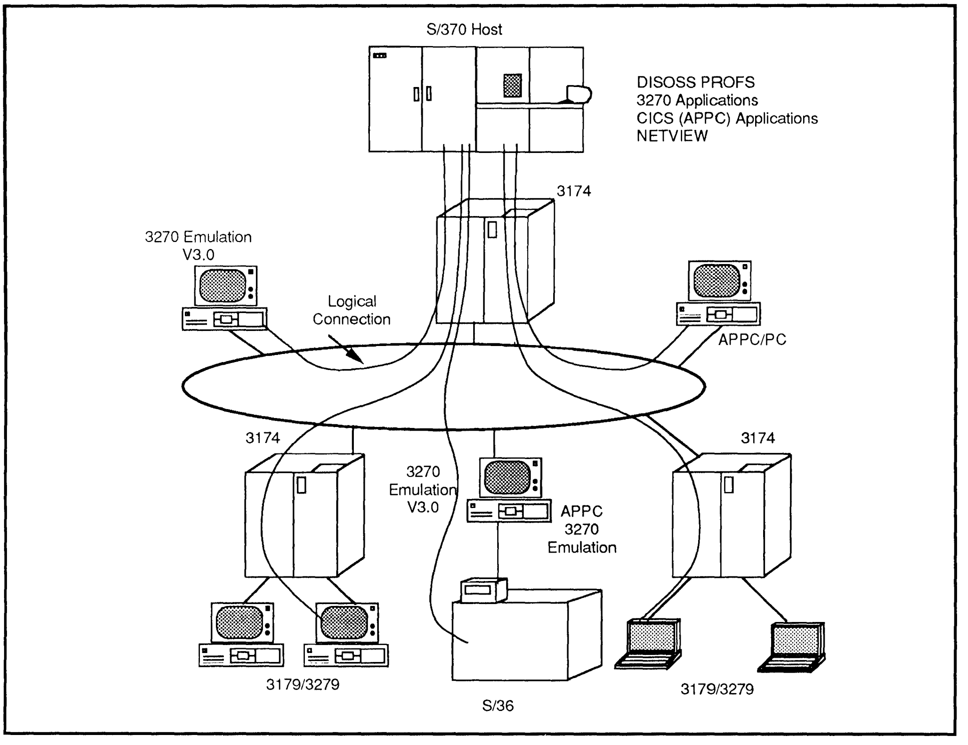

Figure 2-14: 3174 on the Token-Ring

Support-S Release 2 adds Token-Ring Gateway support for remote 3174 Models 1R, 2R, 5R, and SNA/SDLC mode. (The gateway is defined by IBM as 3174 Model 1L, in which other remote 3174 control units that are attached to the Token-Ring may communicate to it -- up to 140 PU2.0 connections on the Token-Ring are supported.) The previous release supported the Token-Ring Gateway option feature with only the 3R and 53R models. It appears that the 3R and 53R continue to be the only models that can also attach locally to a 3720 or 3725 front-end processor.

The 3174 Subsystem Control Unit Type 3A Dual Speed Communication Adapter and supporting microcode, provides both gateway and alternate host attachment for IBM Token-Ring Networks operating at either 16 or 4-Mbps. The 16-Mbps ring can attach to an IBM mainframe, whether it is a backbone or application ring. Substantially improved gateway throughput can be achieved at either speed with the use of larger frame sizes.

3174 Configuration Support-B Release 2 contains functional enhancements to the 3174 including Multi-Host Token-Ring Gateway, which provides enhanced connectivity options for devices attached to the Token-Ring.

The Multi-Host Token-Ring Gateway allows a Token-Ring attached device to take greater advantage of the multiple host connectivity options available in the 3174 Establishment Controller, to access hosts through the primary host link and/or the Concurrent Communication Adapter. With the Multi-Host Token-Ring Gateway function, a user can have sessions with up to three different SNA hosts concurrently through the Token-Ring network using the 3174 configured as a gateway. A maximum of 50 downstream physical units may access the SNA host through each Concurrent Communication Adapter.

Token-Ring-attached devices that require access to multiple hosts via the Multi-Host Token-Ring Gateway will define the path through the 3174 gateway to the specific host link via the SNA Service Access Points portion of the gateway address. This requires definition during the customization of both the Token-Ring-attached device and the 3174 gateway.

Alerts and problem determination statistics from the Token-Ring continue to flow through the 3174 gateway to the SNA host over the primary link. Link events, specific to the sessions in progress via the Concurrent Communication Adapter links, flow to the appropriate host via that link.

3745

The 3745 Communication Controller is the newest member of the IBM Communication Controller family. The 3745, in conjunction with up to four IBM 3746 Expansion Units, offers modular growth for up to 16 host attachments, 512 line attachments, 8 high speed line attachments to T1 and CEPT channels, 8 IBM Token-Ring attachments and provides interfaces for local and remote consoles and the Remote Support Facility.

The 3745 is a medium to high end communications controller which operates under the control of the Advanced Communication Function/Network Control Program Version 5 (ACF/NCP) licensed program.

The 3745 supports IBM's network management direction by sending error related information to the NetView program running in a host processor. Alerts generated for the 3745 are displayed on the network control terminal by NetView, which also provides support for the IBM Token-Ring Network.

The 3745 controls data communications between modem attached, directly attached (without a modem) or IBM Token-Ring Network attached terminal devices, or between such terminal devices and one or more directly or remotely connected IBM 4341, 4361, 4381, 937X, 3033, 308X or 3090 host processors, or between host processors.

It can attach to a byte multiplexer, block multiplexer or selector channel and supports the Data Streaming mode when attached to a block multiplexer channel of an IBM 937X or 3090 system.

The Token-Ring Adapter and the Token-Ring Adapter Type 2 (16-Mbps) features on the same 3745 may be mixed. The Token-Ring Adapter Type 2 supports two attachment ports, software selectable ring speed of 4 or 16-Mbps per port, the use of large I-frames on the ring, INN/BNN traffic on the same Token-Ring Adapter Type 2 port and the Early Token Release option at 16-Mbps. The Early Token Release option allows improved ring utilization particularly on long rings with short I-frame traffic. At 4-Mbps, the Token-Ring Adapter Type 2 has higher data throughput than the current Token-Ring Adapter. Up to four Token-Ring Adapter Type 2 features can be installed on a 3745 allowing attachment to up to eight rings.

The 3745 can support up to 9,999 Token-Ring attached devices operating on 16 or 4-Mbps IBM Token-Ring Networks.

3172

The new IBM 3172 Interconnect Controller is a Micro Channel/80386-based intelligent controller that provides channel attachment of LANs to IBM System/370 host processors. Through multiple LAN attachments, the 3172 provides connection and data transfer services between LANs and IBM System/370 host processors in Transmission Control Protocol/Internet Protocol (TCP/IP) and Manufacturing Automation Protocol (MAP) Version 3.0 networks.

The LANs supported by the 3172 are Token-Ring and IEEE 802.3 (CSMA/CD) which includes Ethernet via TCP/IP protocol and MAP 3.0 (IEEE 802.4 Token Bus) protocol. The data stream is transparent to the 3172. Up to four LAN attachments and two channel attachments can be used with the 3172 Interconnect Controller. The 3172 is available in the IBM 9309 Rack Assembly or may be ordered as a stand-alone unit that can be housed in any compatible EIA 19" rack.

The 9370

Introduced in October 1986, the 9370 Information System offers IBM's widely used System/370 architecture at a "low" entry price relative to other S/370 computers. The 9370 family can connect multiple locations, departments, personal computers, and mainframes.

PCs or PS/2s with the appropriate token-ring adapter may communicate via a token-attachment 9370 Token-Ring when configured with either the IBM PC 3270 Emulation Program Version 3.0 or the IBM 3270 Workstation Program Version 1.1.

Advanced Communication Function/Virtual Telecommunications Access Method (ACF/VTAM) Version 3, Release 1.2 enhancements extend the connectivity support for the 9370's, including an Application Program Interface (API) that has been enhanced to allow APPC between applications using LU6.2 sessions: the applications can run in any IBM APPC support computer. Another enhancement is support for the 9730 Token-Ring Local Area Network Communication Adapter.

VTAM for the VSE operating system supports the IBM Token-Ring Subsystem Controller, allowing a 9370 processor to communicate with another 9370 processor or with terminals on the ring.

APPC/VTAM (LU6.2) is supported for all 9370 operating system environments, including VM, VSE, and MVS/370. This enables System/370 APPC applications to communicate over an SNA network with APPC applications running on other systems. Those systems include another System/370, such as the 9370, System/36, Series/1, System/88, RT PC, an IBM personal computing system, or another manufacturer's processor which supports APPC communications protocols.

IBM has enhanced Teleprocessing Network Simulator (TPNS) Version 2 to Release 4 that supports the simulation of Type 2.1 nodes and PU Type 2 nodes and their resources connected to a token-ring. To perform the terminal simulation, this interface requires the TPNS control program to be executing in a 3725 or 3720 Model 11 with the appropriate token-ring subsystem hardware and microcode. TPNS will simulate one physical address on the ring with the Token-Ring Interface Coupler (TIC) adapter installed, and multiple logical link stations can be simulated per TIC.

Summary

IBM clearly has the hardware to create Token-Ring's that offer connectivity to a wide variety of processors. Administration of the Token-Ring is eased with its structured Cabling System -- management aspects are discussed in Chapter 7. New devices that directly attach to the Token-Ring will continue to be announced by IBM as well as third-party vendors, and the Cable System will evolve yet further. A summary of the evolution in Token-Ring adapter attachments from IBM is illustrated in Figure 2-15.

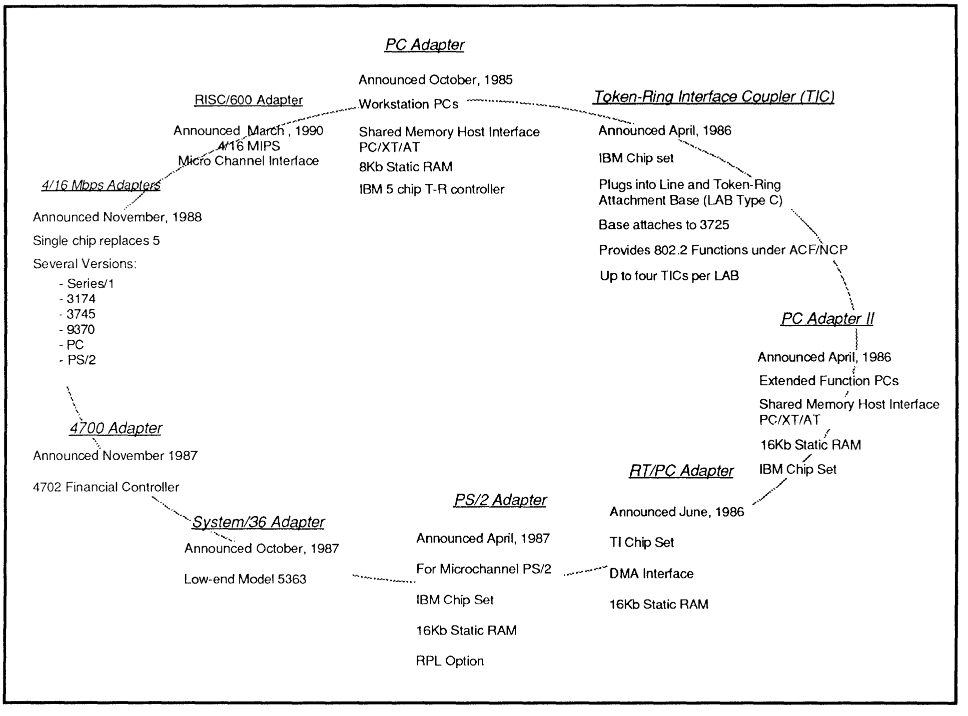

Figure 2-15: IBM Token-Ring Adapter Evolution

Chapter 3

Standards and Protocols

The official token-ring standard is ANSI/IEEE Std 802.5-1985 Token-Ring Access Method and Physical Layer Specifications. This standard has also become the International Standards Organization (ISO) 8802/5 standard.

Since the 1985 standard, there have been several proposed enhancements, most of them as we can imagine, submitted by IBM. As of this writing, none of them have become official standards, but most are in the final draft stage. The additions are as follows:

- 802.5B Recommended Practice for Use of Unshielded Twisted-Pair Cable (UTP) for Token-ring Data at 4 Mbits/sec
- 802.5C Dual Ring Operation with Wrapback Configuration
- 802.5D Enhancement for Multiple Ring Networks (IBM source routing)
- 802.5E Management Entity Specification
- 802.5F Enhancement for 16 Mbits/sec Operation
- 802.5H Recommended Practice for LLC Type 3 Support
- 802.5I Enhancement for Early Token Release
- 802.5L Token-Ring Standard Maintenance (corrections and clarifications in the 1985 802.5 standard)

Protocols and The OSI Reference Model

By definition, a protocol is simply "a set of conventions that allows two or more end points to communicate." Protocols consist of syntax, semantics, and timing.

The syntax of a protocol defines the bit stream (a series of 1's and 0's) by breaking it up into fields. For example, the first 48 bits may be the source address, followed by 48 bits of destination address, 16 bits of packet type, etc. The semantics define the precise meaning of the bits within a field. For example, an address of all 1's may be interpreted to be a broadcast message. Timing is also critical to allow information to flow smoothly over the high-speed local area network. Timing can range from the raw data rate of the bit stream (thus the infamous 10-Mbps Ethernet vs. 4-Mbps token-ring argument) to time-outs between acknowledgements. When discussing the various syntax and semantics of protocols, vendors will often do so with reference to the Open Systems model for Interconnection (OSI) that was developed by the International Standards Organization (ISO). OSI is a model for the connection of non-homogeneous (multi-vendor) equipment in a network environment. Note the wording "network environment" -- the network could be of any classification from X.25-based long-haul nets to intra-city leased-line nets to local area nets. The key is that OSI provides the framework for developing implementable protocols. As a result, many standards committees, such as the IEEE and ISO, have developed protocols that conform to the OSI model (Figure 3-1) that emphasizes peer-to-peer relationships.

The bottom most layer or Layer 1 is called the physical link layer. The OSI reference model states that this layer must specify the encoding technique (such as differential Manchester as used in Ethernet as well as the Token-Ring), the signalling rate, and the interface to Layer 2. The physical layer does not specify the media type and connectors -- after all, layers in the OSI model are independent of the layers above or below and in this case, the physical signalling and bit rate are independent of the media type.

Layer 2, or the data link layer, specifies the data link procedures and packet format. It is this layer that first divides up the bitstream from Layer 1 for field interpretation. Example data link procedures include the IEEE 802.2 Logical Link Control, HDLC, and SDLC. 802.2 is used in the IBM Token-Ring and is broken into Type 1 (connectionless), Type 2 (connection oriented) services, and Type 3 (connectionless with acknowledgement). Type 1 allows sending of datagrams ("one shot" messages) with no prior connection and with no acknowledgements. Type 2 establishes a connection before ac-

LAYERS	*FUNCTION*	*LAYERS*
User Program	Application Programs (not part of the OSI model)	Server Machine
Layer 7 Application	Provides all services directly comprehensible to application programs	Layer 7 Application
Layer 6 Presentation	Restructures data to/from standardized format used within the network	Layer 6 Presentation
Layer 5 Session	Name/address translation, access security, and synchronize & manage data	Layer 5 Session
Layer 4 Transport	Provides transparent, reliable data transfer from end node to end node	Layer 4 Transport
Layer 3 Network	Performs message routing for data transfer between non-adjacent nodes	Layer 3 Network
Layer 2 Data Link	Improves error rate for messages moved between adjacent nodes	Layer 2 Data Link
Layer 1 Physical	Encloses and physically transfers messages between adjacent nodes	Layer 1 Physical

Physical Link

Figure 3-1: OSI Reference Model

tual transfer of data and includes sequence numbers and acknowledgements. Type 3 is Type 1 with acknowledgement.

Layer 3, the network layer, is used to support internetworked environments. It recognizes an additional level of addressing and can be used in a gateway to route packets to other networks.

Layer 4, transport, has the function of supporting reliable communication between two end-points, even on unreliable networks (such as Ethernet that depends on datagrams -- Ethernet systems do not implement IEEE 802.2 Type 2 services). The transport layer adds sequence numbers and acknowledgements and depending on the class of service, error recovery.

Layer 5, or session, allows establishing, maintaining, and breaking down of a session between two end-points, by logical name. NETBIOS is often referred to as a session-level interface, even though it provides other services, including sending and receiving of datagrams (a function of Layer 2).

Layer 6, the presentation layer, contains the encoding rules for data representation (character sets, integer representation, floating point, etc.). For example, Layer 6 may "negotiate" with another Layer 6 residing on another device in a LAN for ASCII character transfer. The other device may be a 3274 gateway in which it will have to perform ASCII to EBCDIC character conversion in order to communicate with the PC. It may have been the other way around in which the PC negotiates for EBCDIC and thus the PC, not the gateway has to perform the conversion. This is obviously an implementation detail and the designer has to study the performance trade-offs. Layer 6 is virtually non- existent in every PC LAN -- ASCII and MS DOS file formats are used by default.

Layer 7, or the application layer, provides network services to applications. Services include file transfer, terminal emulation, logging into a file server, etc. The application layer is often confused with applications as we know them in the PC world -- spreadsheets, wordprocessors, etc. These are really at "Level 8." A real example PC LAN Level 7 application is Novell's NetWare workstation shell software.

In an OSI implementation, each layer communicates by passing its data to the layer beneath it; the message header is appended, and then the data packet is passed to the next layer. This process continues until the physical layer is reached; at that point, the entire encapsulated packet is sent out through the network in the form of a bit stream. When the bit stream reaches the receiving node, it becomes de-encapsulated at the data link level. If the data link level recognizes the data packet, and if the address is correct, it will pass the packet up to the next level. This process continues until the data packet reaches the application level. Although it is possible to send thousands of bytes of data at a time through the physical and data link levels, the actual usable data that is sent is far less due to encapsulation and appending of headers in the process.

ISO has developed protocols for each layer of the OSI model. IEEE has developed protocols for Layers 1 and 2. The IEEE protocols we hear most about are IEEE 802.2, 802.3 (1- or 10-Mbps baseband, bus topology -- StarLAN and Ethernet), 802.4 (1 to 10 Mbps Token Bus, broadband, bus topology -- used in MAP specification), and 802.5 (1- or 4-Mbps, baseband, ring topology -- IBM Token-Ring). The IEEE specifications deviate from the OSI model somewhat as the 802.3, 802.4, and 802.5 standards specify Layer 1 and the lower half of Layer 2. 802.2 is really the upper half of Layer 2.

Other popular protocols such as IBM Systems Network Architecture (SNA), Xerox Network Systems (XNS), and Transport Control/Internet Protocol (TCP/IP) used by the Department of Defense (DoD), were developed before the OSI model. The layering is very similar, however, to the model. SNA provides the functionality of Layers 1 to 7, XNS covers Layers 3 to 7, and TCP/IP covers Layers 3 and 4.

IEEE 802.5 Media Access Control

The following coverage of IEEE 802.5 and 802.2 gives a flavor for the operation of Media Access Control (MAC), Physical (PHY), Logical Link Control (LLC). For more detailed information, consult the IEEE 802.2 and IEEE 802.5 specifications as referenced in Appendix A.

IBM's contributions to the IEEE 802.5 standards committee forecasted the formal introduction of IBM's Token-Ring products. Obviously, IBM has been the most influential company in shaping the 802.5 standard.

Physical Layer

In the token-ring, the physical layer (PHY) is capable of transmitting and detecting four different symbols: a binary 0, a binary 1, a non-data-J, and a non-data-K. Differential Manchester type coding is used. The J and K symbols are "violations" of Manchester coding used to denote the beginning and ending of a frame. An example of the symbols is shown in Figure 3-2.

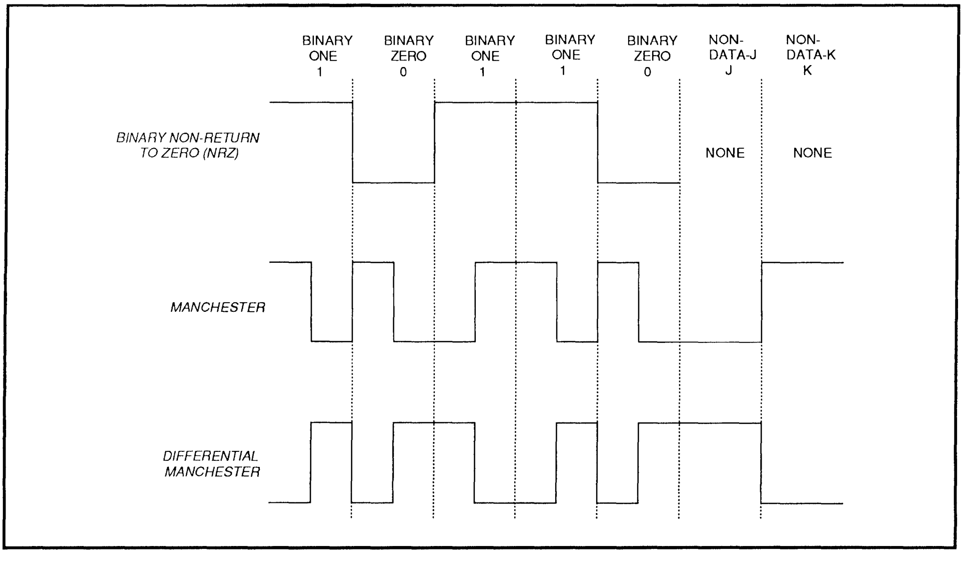

Figure 3-2: Manchester Encoding

The 802.5 transmission rate can be 1- or 4-Mbps with a tolerance of +/- 0.01 percent. Since the release of the 16-Mbps IBM Token-Ring, a proposal is out to drop the 1 Mbps rate (no one has implemented it) and to add the 16-Mbps rate (the electrical details are spelled out in the document "IEEE 802.5F Enhancement for 16 Mbits/sec Operation").

A latency buffer is provided by the active monitor. In ensures enough latency time (bit delay) is in the ring to allow the token to continuously circulate around the ring. This bit delay is the token size, namely 24 bits, plus some additional bits (detailed later). The actual bit delay in the ring will depend on the number of attached devices (roughly 2 bits per station) and the velocity of propagation times the total length of active wire in the ring.

The 4- or 16-Mbps timing is provided by the active monitor. All other stations derive their timing by tracking the frequency and phase of the incoming signal. Though the mean data signalling rate around the ring is controlled by the monitor, segments of the ring can instantaneously operate at speeds slightly higher or lower than the frequency of the monitor. The cumulative effect of these variations in speed may cause an effective variation of up to +/- 3 bits for 4-Mbps operation or +/- 16 bits for 16-Mbps operation in the latency of a ring that has been configured with a maximum number of stations (250 for 802.5, 260 for IBM -- both using the data grade wire as discussed in Chapter 2). To compensate for these variations, the active monitor's latency buffer can expand to 30/56 bits (4/16-Mbps) for a "fast" signal, to avoid dropping bits or contract to the token size of 24 bits for a "slow" signal, to avoid adding bits to the repeated bit stream.

The IEEE 802.5 standard contains the station attachment specification for shielded twisted-pair and an accompanying medium interface connector.

A detail of the station connection to the medium is shown in Figure 3-3. Station insertion into the ring is controlled by the station. The station places a dc voltage on the media interface cable that is transparent to the normal operation (Manchester coding of symbols) of the ring; hence it is called a "phantom" voltage. This voltage causes switching of a relay in a Trunk Coupling Unit (also known in IBM terminology as a Multistation Access Unit), resulting in the serial insertion of the station into the ring. A loss of the phantom voltage causes a relay to bypass the station and in turn, causes the station to put in a looped (wrapped) state. This looping feature is used by IBM adapters as part of the diagnostics performed when the adapter card when powered on.

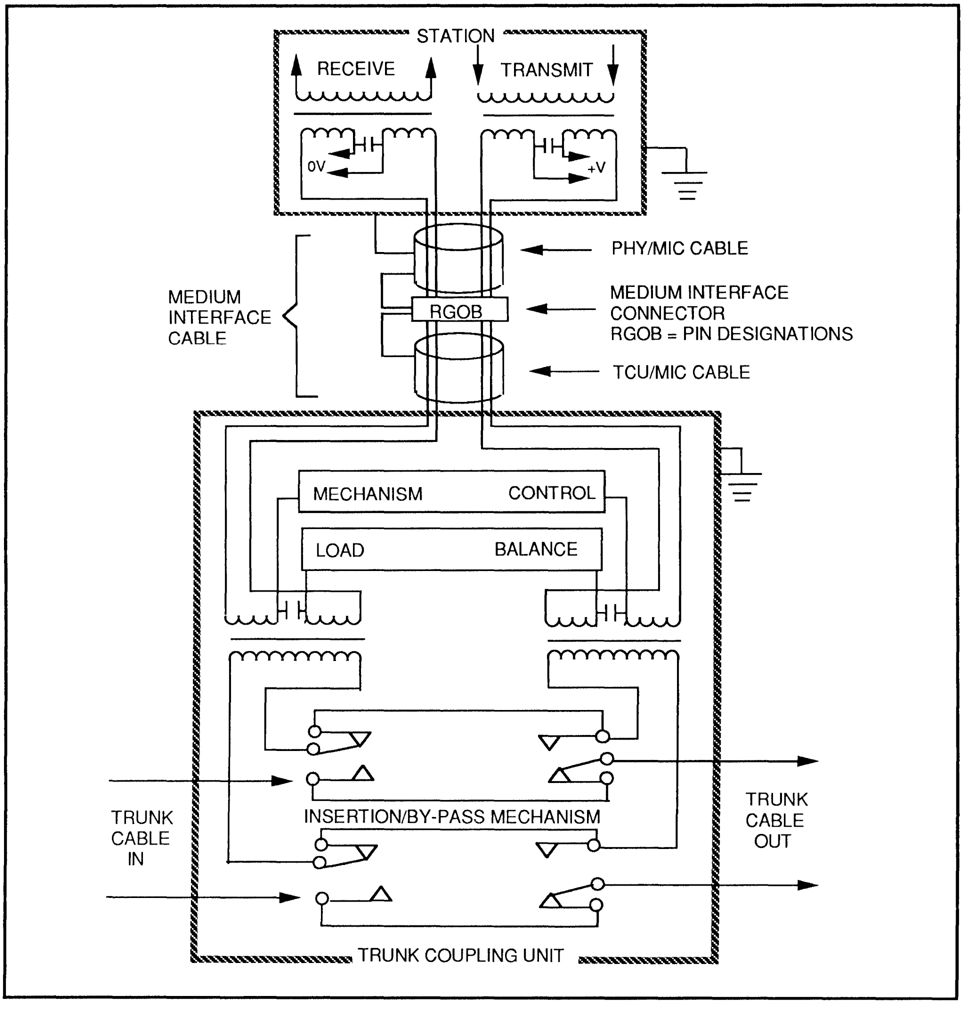

Figure 3-3: Physical Medium Connection Detail

Figure 3-4 depicts the medium interface connector (MIC), also known as the IBM data connector. The MIC contains four contacts with a ground contact and is hermaphroditic in design. Two identical connectors will mate when oriented 180 degrees with respect to each other.

The MAC Protocol

802.5 defines the medium access control (MAC) sublayer of the data link layer in addition to the physical access layer. This partitioning

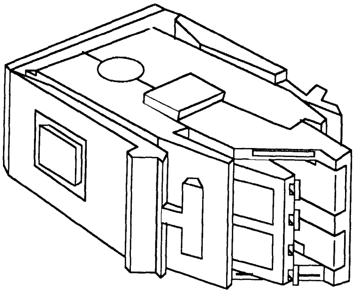

Figure 3-4: Medium Interface Connector

is shown in Figure 3-5. The frame format is defined, including delimiters, addressing, and frame check sequences. Also defined are medium access control frames, timers, and priority stacks. At the physical layer, the functions of symbol encoding and decoding, symbol timing, and latency buffering are defined.

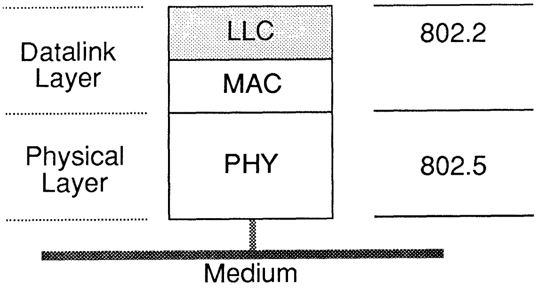

Figure 3-5: IEEE 802.2/802.5 Partitioning

802.5 describes services provided by the medium access control sublayer to the network management and the logical link control sublayer and the services provided by the physical layer to network management and the medium access control sublayer. These services are defined in terms of service primitives and associated parameters, much the same way 802.2 is defined.

A token-ring consists of a set of stations connected serially by a transmission medium in which information is transferred bit by bit from one active station to the next. A station regenerates and repeats each bit, thus acting as a repeater when active. According to 802.5, a station is an entity which "serves as the means for attaching one or more devices (terminals, workstations) to the ring for the purpose of communicating with other devices on the network."

A station having access to the medium transfers information onto the ring where it circulates from one station to the next. A station recognizing its address, copies the information as it passes "through" the interface. The information transmitting station is responsible for removing the information from the ring, preventing further circulation.

When a token (a unique sequence of bits that circulates on the ring) is detected by a station, it may transmit its information onto the medium. When a token is "captured" by a station, the station modifies the start-of-frame sequence (the first two bytes of the three byte token) and appends the appropriate control and status fields, address fields, information field, frame-check sequence and the end-of-frame (the last byte of the original token) sequence.

To aid in fair access to the ring, a token holding timer controls the maximum time a station can use the ring before passing the token. At the end of the information transfer, the station checks to see if its frame header (the first three bytes of the frame) has been receive after traversing the ring; if not, the stations continues to transmit idle bits until the header is returned. At this time, the station will generate a new token.

A variation on this was introduced with IBM's 16-Mbps token-ring called "early token release" (proposed as an 802.5 standard entitled "802.5I Enhancement for Early Token Release). With early token release, the station does not have to wait for its header to return. Instead, it can release the token at the end of the transmission of a frame, if its header has not returned.

Multiple levels of priority are available depending on the relative class of service required such as synchronous (3270-type data streams) and asynchronous (interactive) data transmission. An immediate network recovery has the highest priority. The allocation of priorities is done among the stations active on the ring.

One innovative aspect of the 802.5 specification is built-in error detection and recovery mechanisms provided to restore network operation in the event of failed medium or medium transients (insertion and removal of stations). Detection and recovery utilizes a network monitoring function that is performed in a specific station called the "active monitor" with back-up monitoring capability by other stations in the network. Other stations also participate at all times in checking for errors by performing a CRC on each frame as it passes through that adapter of that station.

Token and Frame Formats

The MAC formats can be divided into two types: token and frames. In the following figures, the formats (generally described in terms of octets -- 8-bit fields) are depicted in the sequence in which they are transmitted on the medium, with the left-most bit transmitted first. The left-most bit is also considered to be the most significant bit.

The token, as shown in Figure 3-6, is the mechanism by which the right to transmit on the ring is passed from one station to another. It consists of three bytes (or octets) containing the starting delimiter (SD), access control (AC) and ending delimiter (ED). These three bytes are also used in the frame format (Figure 3-7). The frame format is used for transmitting both MAC frames and logical link control (LLC) messages to the destination station (or stations if the destination is a group or broadcast address). Between the access control and ending delimiter resides the information field (often called an I-field) and a 32-bit CRC frame check sequence. The I-field is a "bit stream" in the sense that it can be defined as bits, bytes, nibbles (4 bits), words, etc. The maximum frame size is 2 Kbytes for the 4-Mbps IBM Token-Ring and 18 Kbytes for the 16-Mbps IBM Token-Ring.

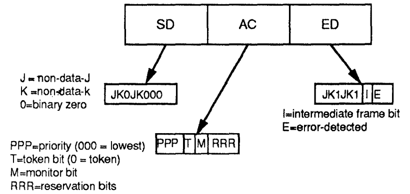

Figure 3-6: Token Format

The starting delimiter uses non-data J and K symbols (those which are not valid binary bits) to indicate the start of a token or frame. This avoids the overhead encountered in protocols like SDLC (which uses 01111110 as frame delimiters). Since this pattern serves a unique purpose for SDLC, it cannot be in the frame itself and thus bit stuffing and removal must be performed.

Likewise, the ending delimiter also uses the non data JK symbols in a unique way to indicate the end of a token or frame.

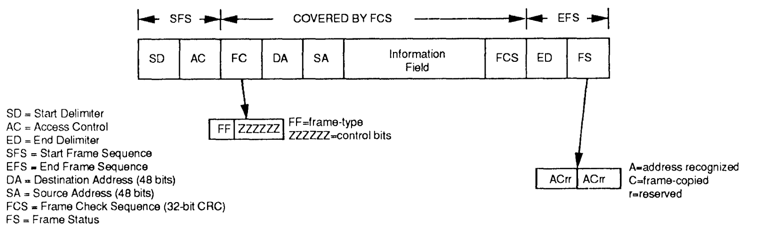

Figure 3-7: Frame Format

The access control field contains priority bits, (3 bits - PPP) a token bit, a monitor bit, and reservation bits (3 bits - RRR). The lowest priority is 000, the highest 111. A token bit of 1 indicates a frame follows. When a station wishes to transmit a protocol data unit (PDU) and detects a token with a priority equal to or less than the stations priority bits, it may change the token to a start-of-frame sequence and send the PDU. The monitor bit prevents a token with priority greater than 0 or a frame from continuously circulating on the ring. The bit is transmitted as 0 in all frames and tokens except for the monitor which inspects and modifies it.

Reservation bits allow stations to request the next token be issued at the requested priority. The reservation bits can be set while a frame passes through a station. The station currently transmitting the frame is responsible for examining the reservation bits when the frame header returns and if necessary, generate a new token at a higher priority (the PPP bits are set) and for changing the token back to the lower priority when the requesting station has completed frame transmission.

The frame control field indicates frame type, along with control bits. FF set to 00 indicates a MAC frame; 01 indicates an LLC frame; 1x is an undefined format reserved for future use.

Each frame contains destination and source address (DA and SA) fields. 802.5 allows field size to be either two (16 bits) or six (48 bits) octets. The IBM Token-Ring uses six octets for addressing plus optional source routing information. The DA and SA format is illustrated Figure 3-8. If the first bit is set to 0, then it is an individual address. If set to 1, then it is a group or broadcast (all bits set to 1) address.

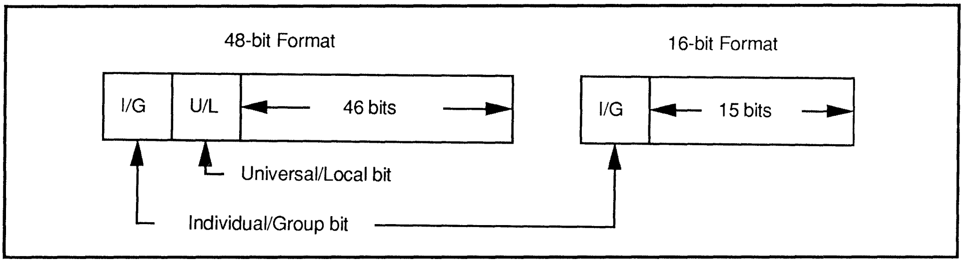

Figure 3-8: Source and Destination Field Formats

The second bit in the address field indicates if the address is locally (set to 1) administered or universally (set to 0) administered. A universal address is preassigned to each station by the manufacturer (the manufacturer can obtain its own set of 48-bit addresses from the IEEE, formerly administered by Xerox for Ethernet LANs), and is unique across the universe of LANs.

The frame status field contains address recognized bits (A bits), frame copied bits (C bits), and reserved bits. A and C bits are transmitted as 0 by the originating station. When the receiving station recognizes its address, it will set the A bits to 1. If it was able to copy the frame into its receive buffer, then the C bits are set to 1. This allows the sending station to determine if the station is non-existent or non-active on the ring, if the station is active but is unable to copy the frame, or if the station is active and did copy the frame.

The MAC frame itself consists of vectors and subvectors. Figure 3-9 illustrates the possible frame formats as defined by IBM in the IBM Token-Ring Architecture Reference. The major vector ID vector (MVID) identifies the class and code of the MAC frame; the subvectors supply further detailed information. For example, a MVID of 03 hex is a claim token frame. The MAC frames defined by the 1985 IEEE 802.5 standard include: claim token, duplicate address test, active monitor present, standby monitor present, beacon, and purge. The additional MAC frames have been defined by IBM to aid in management and problem determination procedures. In the interest of compatibility, these additional MAC frames are implemented by token-ring chip vendors and have been submitted for standardization to IEEE. The following section contains additional detail on the type of information required for these (management) MAC frames.

Management Entity

The proposed IEEE 802.5E Management Entity Specification specifies the services provided by MAC to LLC, PHY to MAC, MAC to SMT (station management) and PHY to SMT. The services are

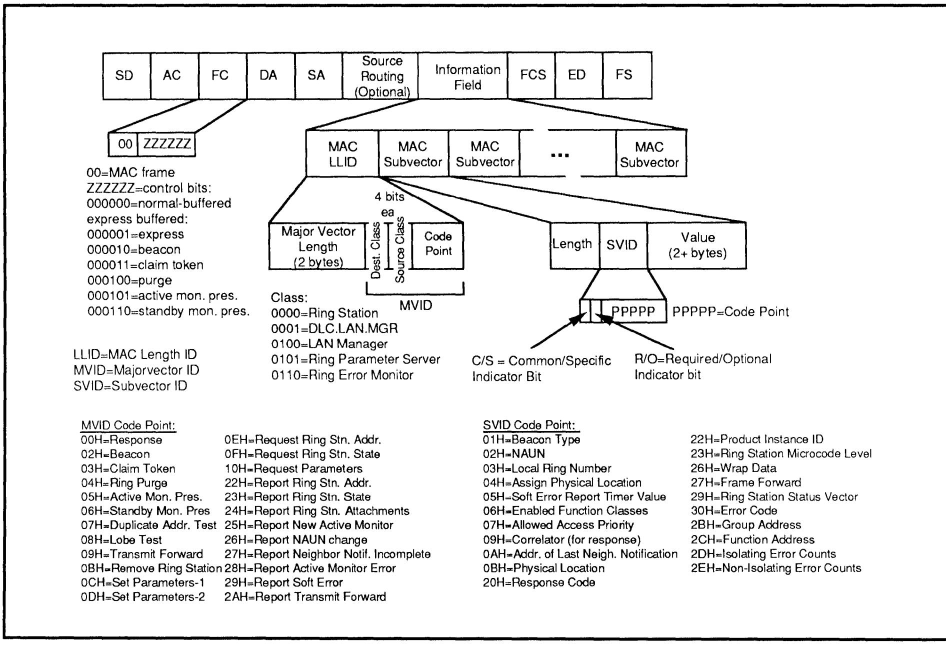

Figure 3-9: IBM MAC Frame Formats

specified in an abstract way and do not imply any particular implementation or interface.

MAC to LLC

The MAC to LLC service specifies the services required of the MAC by LLC to allow exchange of data units with peer LLC entities. Work is in progress to define a common specification for all MACs, not just the token-ring. MAC specifies primitives for REQUEST (transfer of MAC data from local LLC to peer LLC) and indication (transfer of data from MAC entity to LLC entity, or reception of data). Specified in either case is the destination address, source address, data, priority (up to eight levels), and service class (class of service desired for the data unit transfer).

PHY to MAC

The PHY to MAC allows data exchange between peer MAC entities. This interaction is handled entirely by hardware (i.e. the token-ring chip set). Data that is transfered includes binary ones and zeros as well as the non-data-J and non-data-K symbol required by the token-ring.

MAC to SMT (station management)

MAC to SMT specifies parameters, events and actions between MAC and SMT which consist of reporting errors, parameter values, and other events. This is a requirement for token-ring implementations and all token-ring chip sets implement this requirement.

Information available to SMT from MAC includes address group, attachment group, status, statistics, transients (events and error reporting).

Address group reports the following information: station MAC address, functional addresses, group MAC address, upstream neighbor address, ring number, physical drop (4 bytes -- installation dependent), and a private address parameter (which allows a vendor specific address group parameter).

Attachment group reports the following information: functional addresses, authorized functional class (Ring Station, Configuration Report Server, Ring Parameter Server, or Ring Error Monitor), authorized access priority, product instance ID, and private attachment parameter.

Status identifies the current state of the resource. Included are the MAC version numbers, MAC status, and private state parameter.

Statistics include isolating error counters (that can be used to isolate a fault domain (between two stations) on the ring) and non-isolating error counters. Isolating error counts include line errors (such as a CRC error), burst errors (an absence of transitions on the media), ac errors (active monitor problem), abort transmission error, internal error, and private errors (vendor specific). Non-isolating error counts include lost frames (station transmits but doesn't receive its frame back), receive congestion (station recognizes a frame addressed to it but has no buffer space for the frame), frame copied error (a frame addressed to that station already had the frame copied bit set by some other station), token error (the active monitor sees a condition that needs a token to be transmitted), and private errors.

Transients concern the dynamic actions of a resource (actions and event reports). Actions include Disconnect (remove from ring), Open (insert into ring), and private action (vendor defined). Reports are generated automatically or by request (from an incoming MAC frame from another stations). The information contained in this reports is as follows:

Event reports include enter active state, active monitor error, report station in ring, configuration change, neighbor notification incomplete, counter threshold reached, beaconing condition on ring, ring station removed, and private event (vendor defined).

The Enter Active State Report (generated by request from a Configuration Report Server)) includes the following information: ring number, active monitor address, upstream neighbor address, physical drop, and product instance ID.

The Active Monitor Report (generated when the Ring Error Monitor receives a report active monitor error from the new active monitor) includes ring number, station MAC address, upstream neighbor address, error code (active monitor error, duplicate active monitor, or duplicate address), and physical drop.

Report Station in Ring Report (generated when the Ring Parameter Server receives a request initialization message from a station when the station has inserted and is about to operate in the ring) includes the ring number, station MAC address, upstream neighbor address, product instance ID, and MAC version number).

The Configuration Change Report (generated when the Configuration Report Server receives a Report SUA (station upstream address) Change message from a station) includes the ring number, station MAC address, upstream neighbor address, and physical drop).

The Neighbor Notification Incomplete Report (generated with the Ring Parameter Server receives a Report Neighbor Notification Incomplete message from the active monitor) includes the ring number, active monitor MAC address, and source address of the last Active Monitor Present or Standby Monitor Present frame.

The Counter Threshold Reached Report (generated with a threshold is reached or the counter has reached its maximum value) includes the ring number, station MAC address, upstream neighbor address, the isolating error counter values, the non-isolating error counter values, and the physical drop.

The Beaconing Condition on Ring report (generated by the Ring Error Monitor with its attached ring has beaconed or is beaconing) contains the ring number, station MAC address, upstream neighbor address, beacon type, ring status, and physical drop. The ring status can be one of: normal, temporary beaconing (the ring recovered), and permanent beaconing (the ring did not recover and manual intervention is required).

Finally, the Ring Station Removed Report (generated by the Configuration Report Server when it issues a Remove Ring Station frame and verifies that the station is removed) contains the ring number and station MAC address of the removed station.

PHY to SMT

PHY to SMT service specifies abstract entities actions for the PHY to SMT interface. These actions include remove from ring, insert into ring, and private action (vendor defined).

Using Token-Ring Management Information

We can see that by using the information provided by the management entity of the token-ring, we can create a powerful token-ring management and control program (that incorporate features of the "Servers" mentioned such as the Configuration Report Server and Ring Parameter Server). Vendors including Madge and IBM offer programs that operate in PCs and take advantage of a number of these features. For example, IBM bridges can act as a ring parameter servers. The IBM LAN Manager (see chapter 7) can act as Ring Error Monitor and Configuration Server and offer additional features. One such feature is the ability to watch for a "Report Station in Ring Report", compare the station MAC address with a list of authorized stations, and send a Disconnect if the station is not authorized to be on the ring!

Source Routing

Source routing (a proposed IEEE 802.5 standard entitled "802.5D -- Enhancement for Multiple Ring Networks) is an enhancement IBM made to the original IEEE 802.5 specification to allow routing of frames between a multiple-ring network linked by bridges (also called MAC-relay stations). Unlike routing performed by layer 3 (OSI network level) gateways, source routing does not require routing tables to reside in the gateway; each frame carries information about the route it is to follow. The routing information is acquired through a search that originates at the source station that requires an all-rings broadcast using a TEST or Exchange Identification (XID) Logical Protocol Data Unit (LPDU) command.

As the broadcast frame fans out through a multiple-ring network, copies are created by bridges. The frame will eventually reach the destination address in which the destination station returns the acquired routing information back to the sender.

Note that using this technique may result in more than one frame being created for the destination and thus returned to the sender. It is up to the origination station to select which one to use for the route. The algorithm may be as simple as the first frame returned which may be the shortest route provided the intervening networks were not very busy at the time. The optional source routing information is illustrated in Figure 3-10 which can add up to 8 two-byte segment/bridge numbers to each frame.

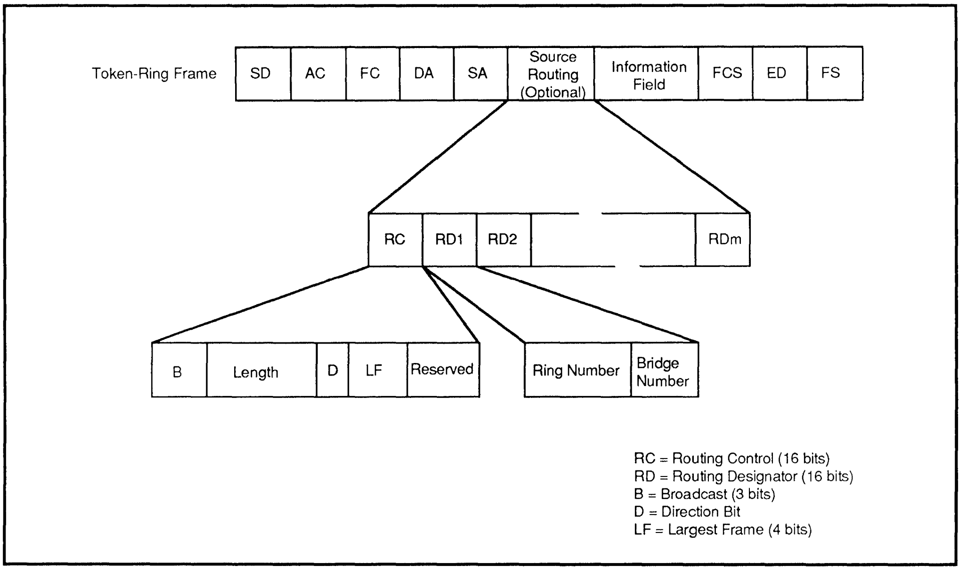

Figure 3-10: Source Routing Field

A bridge or station determines whether or not the frame contains routing by examining the most significant bit in the source address with a 1 indicating that routing information is present. Normally this bit in the source address field is "reserved", since the same field in the destination address indicates group or broadcast addresses (all source addresses are individual addresses). IBM's use of this "reserved bit" will appear that it will become an standard way of indicating source routing -- at least for token-ring (there is no technical reason why this technique couldn't be used in other LANs as well since vendors and users don't want to change the millions of Ethernets already out there!). Eight segment numbers allows up to seven intervening bridges between LANs. There can be more than seven bridges total, since some of them may operate in parallel or a single ring may function as a backbone for dozens of rings.

Error Detection and Correction

IBM stresses the reliability features of the Token-Ring in terms of its Reliability, Availability, and Serviceability (RAS) features. These features are largely built-in token-ring management entities as previously noted.

When a failure is detected in the ring, its cause must be isolated (with the help of diagnostics or management programs) to the proper domain so recovery actions can occur. The domain consists of the station reporting the failure (beaconing station), the upstream neighbor of the beaconing station, and the ring medium between them.

A high level of error detection exists along the transmission path and the active ring elements (devices). The two major classifications of possible errors are soft (intermittent) errors and hard (disruptive) errors.

Soft errors are intermittent disruptions of the ring's signal path resulting in performance degradation (the ring can still operate). Examples of actions causing soft errors are: inserting and de-inserting stations; lost or multiple monitors; line errors (from wrong cable selection, size, distance, or interference from transformers, fluorescent lights, etc.); lost or bad delimiters; multiple or corrupted tokens; lost tokens or frames; and continuously circulating frames (without tokens).

Hard errors result in complete disruption of the ring's signal path preventing normal ring operation. Examples of actions causing hard errors are: failures in the device's adapter such as the transmitter, receiver, repeater logic, microprocessor, or active monitor frequency shift (bad crystal); and failures in the signal path such as broken wires or connectors, or a MAU failure. Broken lobe wiring usually results in a soft error with the station removing itself from the ring. The worst case broken wire scenario is one in which the cable is cut between MAUs. In this case, the ring is disrupted until the wire is manually repaired or pulled from the system (caused a wrap back in the MAUs).

To recover from errors, MAC-level ring recovery protocols come into play to perform automatic switching out of bad devices, perform reconfiguration, and report error events. Recovering from hard errors such as a cable break between MAUs, may require additional manual intervention to isolate cable or connector faults.

Internal diagnostics in all IBM Token-Ring adapters test the internal circuitry, ring interface drivers and receivers, and the external cable and lobe wiring.

All token-ring adapters go through six basic phases to become active on the ring. The first phase involves checking the lobe wiring to the MAU and back via a wrap mode. The second phase is checking for an active monitor; if one is not detected, that PC will become an active monitor. The third phase is to make sure a duplicate address does not exist on the ring. The fourth phrase is neighbor notification, where all nodes become aware of its upstream neighbor's address. The fifth is to request parameters (such as ring number) from a network manager or bridge program (defaults are used if a manager or bridge is not found). Finally, the adapter waits for an active monitor present frame and then becomes an active node on the ring, passing tokens and frames.

While operational, an adapter detects errors on the signal path by loss of signal, burst error detection, signal code violation checking, 32-bit checksum in frame, and MAC frame validation.

The active monitor plays a critical role in monitoring the ring for proper operation and initiating error recovery procedures. It monitors for a lost token (its "any token timer" expires), circulating frames, and circulating priority tokens (to prevent "hogging" by high-priority devices). Furthermore, the monitor provides the master oscillator to clock data at 4- or 16-Mbps. Clocking information at the other active devices is actually derived from the Manchester signalling on the wire. In addition, the monitor provides the latency buffer as discussed previously.

Another duty of the active monitor is to initiate (every seven seconds) the neighbor notification procedure enabling stations on the ring to obtain the address of its next active upstream neighbor (NAUN). This procedure is illustrated in Figure 3-11, and is required by an inserting station before it can begin normal ring operation.

Token-claiming is used to elect an active monitor should the current active monitor fail. This process starts if one of the three following conditions occurs: 1) The active monitor detects a loss of signal, has a expired receive notification counter, or cannot receive enough of its own Ring Purge MAC frames; 2) A standby monitor detects a loss of signal or expiration of its good token counter or receive notification counter; 3) A station inserts into the ring and does not receive the active monitor present frame.

The station which detects the error will start broadcasting claim token frames. As a claim token frame traverses the ring, each station will examine the frame and repeat the frame, or send its own claim token frame is its address is higher that the source address in the current claim token frame. Eventually, the frame that returns to the station with the same source addresse as that station will cause that station to issue a purge ring frame and go into the role as an active monitor. Figure 3-12 summarizes this procedure.

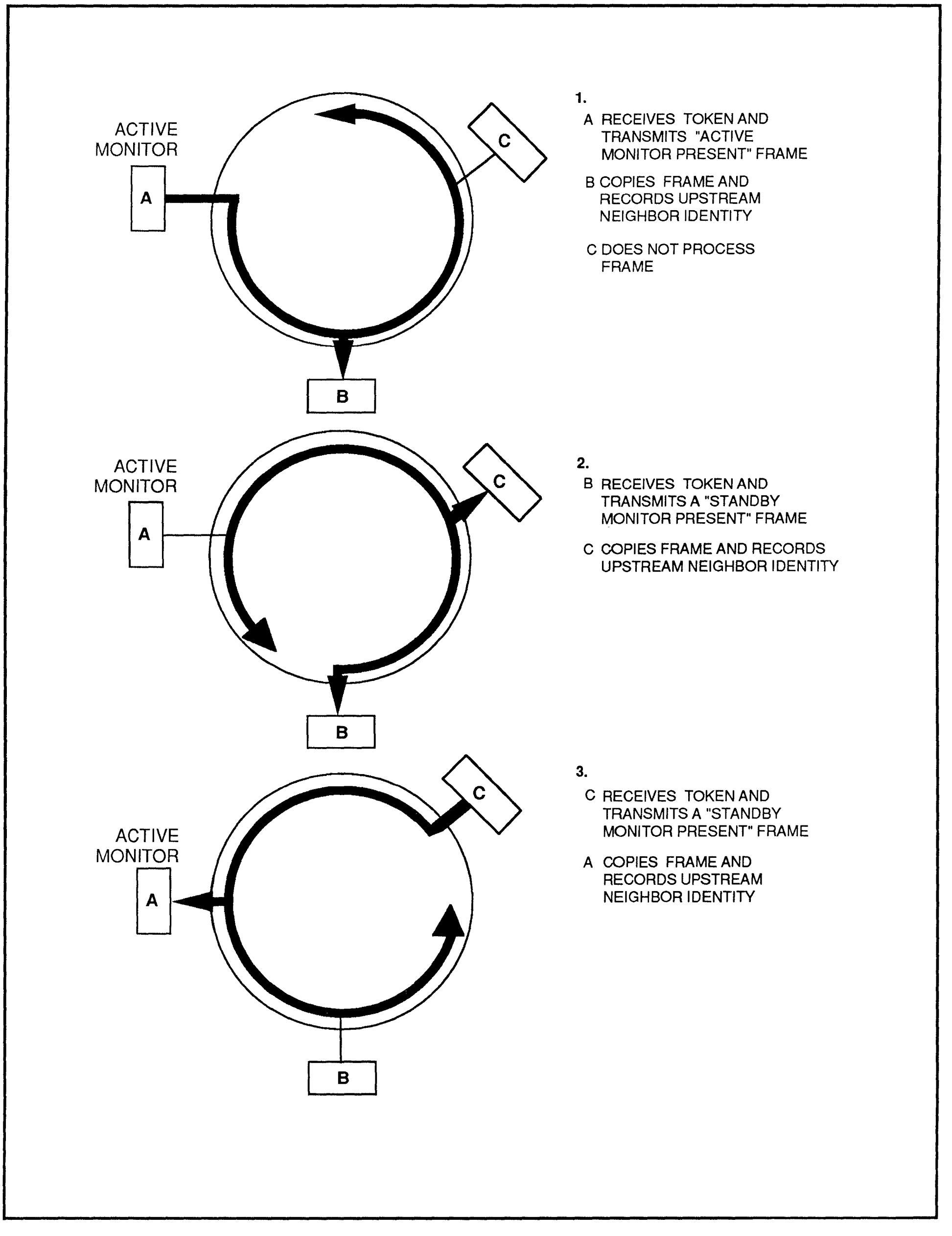

Figure 3-11: Token-Ring Neighbor Notification

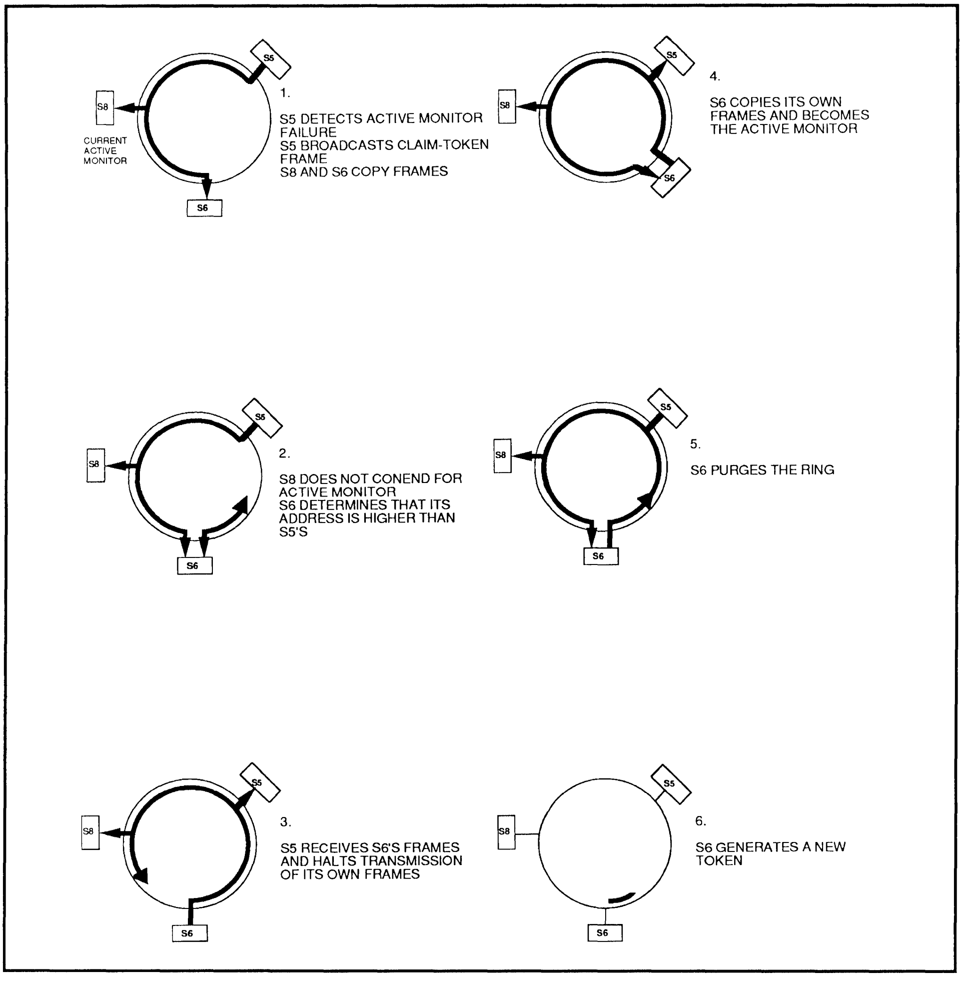

Figure 3-12: Token Claiming

If an individual station detects an error, it may send out a beacon frame to aid in recovery. The beacon identifies itself and its upstream neighbor (NAUN). If the frame makes it to the station with the NAUN address in the beacon frame, it will remove itself from the ring and perform diagnostics tests to determine if reinsertion is possible. If failure continues, then the station which initiated the beaconing station will remove itself (after receiving eight of its own beacon frames) from the ring and perform a diagnostics test to see if it can be re-inserted. These procedures ensure that the offending station or stations attached to defective lobe wiring removes itself from operation. The amount of time needed to recover from a soft error depends on the severity of the error.

If automatic recovery fails, then manual procedures must be initiated to restore operation to that device or perhaps a portion of the ring. To assist in recovery, a program similar to the IBM LAN Manager should be used to display and isolate errors. Once the fault domain is determined, the administrator uses problem determination procedures to perform one or more of the following operations: remove the station with the NAUN from the ring, bypass the NAUN's access unit, remove the beaconing adapter from the ring, bypass the beaconing adapter's MAU, and bypass or replace cables between MAUs. These operations will result in a bypassed MAU, a station's lobe wire removed (unplugged from the MAU), or a disconnected or replaced cable.

If soft errors are persistent, users will notice ring degradation. A PC-based ring management program can be activated to determine the adapter with the highest error count, bypass that adapter's access unit, and bypass or replace cables between access units.

If the above procedures do not isolate the offending cable or device, further steps may be taken. In the case of a single wiring closet, the user can create a test ring consisting of a single MAU. MAUs are added to the test ring until the ring fails. In this manner, a failed MAU can be isolated. In the case of multiple wiring closets, the wiring closets can be isolated from each other, followed by isolation of the MAUs within the closet. The ring test is performed connecting another MAU each time, then to other wiring closets, and its MAUs, until the offending MAU is isolated.

Dual Ring Operation

There is a new proposal submitted to IEEE 802.5 for fault tolerant ring operation known as 802.5C, Dual Ring Operation with Wrapback Configuration. The essence of the proposal standard is summarized here.

This will be an option for token-rings to enable faster recovery from station and/or media failures. Applicable applications include real time systems, systems with high availability requirements, and high integrity systems.

The proposal suggests that dual ring stations with wrapback be an option to the existing 802.5 standard and that non-reconfiguring stations be able to coexist on the same system's primary ring.

The system consists of two counter-rotating rings with only the primary ring operating until the back-up (secondary) ring is needed. Dual ring stations are connected to neighboring dual ring stations with two Multistation Access Units (MSAUs or MAUs -- also referred to as Trunk Coupling Units in the IEEE 802.5 standard). Each dual ring station will contain two 802.5 adapters and a crosspoint switching function controlled by a dual ring management function.

The dual reconfiguration employs a wrapback feature to provide for recovery from all forms of signal loss and from lobe wiring failures not covered by the single ring standard. The wrapback feature works by interconnecting the two rings on each side of the failure, resulting in a single ring being formed from the remaining "good parts" of the original two rings. This concept is illustrated in Figure 3-13.

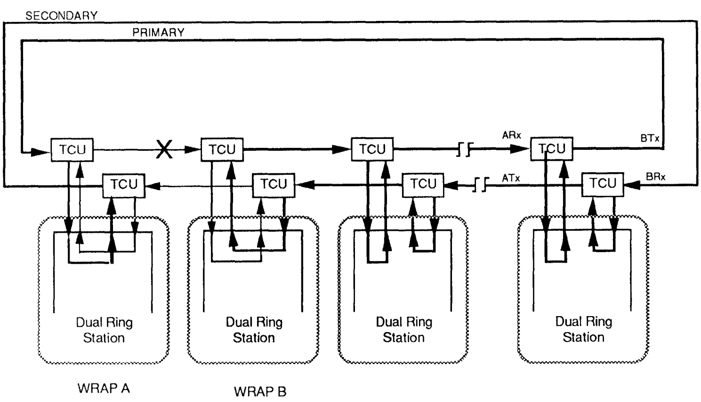

Figure 3-13: Dual Ring Failure Recovery

IEEE 802.2 Logical Link Control

All IBM Token-Ring adapters are capable of executing functions of the logical link control (LLC) sublayer of the IEEE/Std 802 local area network protocol. The LLC sublayer is the "upper" layer of the data link layer (refer back to Figure 3-5). The idea of LLC is to be independent of the network dependent (MAC) sublayer.

LLC sublayer interface service specifications to the network layer (Layer 3), to the MAC sublayer, and to the LLC sublayer management function are described in the LLC Standard. The standard provides a description of peer-to-peer protocol procedures that are defined for the transfer of information and control between any pair of data link layer service access points on a LAN. Two "classes" of LLC operation are provided. Class I provides connectionless service. Class II provides both connectionless and connection- oriented service. The IBM token handler (discussed in Chapter 4) provides an interface to class II service.

Services provided by a layer are specified by description of the service primitives and parameters which characterize each service. Primitives are of three generic types: 1) request -- a primitive passed from the user at layer n (called n-user by IEEE) to layer n-1 (called n-layer by IEEE) requesting a service be initiated; 2) indication passed from layer n-1 and significant to layer-n -- this event may be logically related to a remote service request, or may be caused by an indication internal to the layer n-1; 3) confirm -- passed from layer n-1 to layer n to convey the results of one or more associated previous service request(s).

Connectionless Services

Connectionless service is the data transfer service providing means for network entities to exchange link service data units (called LSDUs) without the establishment of a data link level connection. The data transfer can be point-to-point, multi-cast (group), or broadcast (to all entities).

Connection-Oriented Services

Connection service provides the means for establishing, using, resetting, and terminating data link layer connections. These connections are point-to-point connections between link layer service access points (LSAPs). The service provides data link layer sequencing, flow control, and error recovery. Other services provided include: connection reset (established connections can be returned to the initial state), connection termination (a network entity can request or be notified of data link layer connection termination), and connection flow service (provides flow control of data associated with a specified connection).

The following provides a quick overview of the services provided by the LLC.

LLC Service Primitives

The following is a listing of the service primitives defined by 802.2 LLC. The actually implementation and interface to such services is determined by the implementor (vendor) of 802.2.

<u>L_DATA.request</u> - passed to the LLC sublayer to request a link layer service data unit (LSDU) be sent. The semantics are: L_DATA.request(local_address, remote_address, l_sdu, service_class).

L_DATA.indication - passed from the LLC sublayer to indicate the arrival of an LSDU. The semantics is the same as L_DATA.request.

L_CONNECT.request - passed to the LLC sublayer to request that a logical link connection be established between a local link layer service access point (LSAP) and a remote LSAP. The semantics is: L_CONNECT.request (local_address, remote_address, service_class).

L_CONNECT.indication - passed from the LLC sublayer to indicate the results of an attempt by a remote entity to establish a connection to a local LSAP.L_CONNECT. The semantics is: request (local_address, remote_address, status, service_class).

L_CONNECT.confirm - passed from the LLC sublayer to convey the results of the previous associated L_CONNECT.request primitive. The semantics is the same as L_CONNECT.indication.

L_DATA_CONNECT.request - passed to the LLC sublayer to request an LSDU be sent using connection-oriented procedures. The semantics is: L_DATA_CONNECT.request(local_address, remote address, lsdu).

L_DATA_CONNECT.indication - passed from the LLC sublayer to indicate the arrival of an LSDU. The semantics is the same as L_DATA_CONNECT.request.

L_DATA_CONNECT.confirm - passed by the LLC sublayer to convey the results of the previously associated L_DATA_CONNECT.request primitive. The semantics is: L_DATA_ CONNECT. confirm(local_address, remote address, status).

LLC Packet Format

Figure 3-14 illustrates the packet format used by the LLC protocols -- officially known as the LLC protocol data unit (PDU). The IEEE 802.2 specification defines the method for representing the data link layer service access point (SAP) address to or from network layer entities. IEEE 802.2 also defines a partition of these service access point addresses into individual and group addresses as illustrated in Figure 3-14. IBM has reserved the SAP value F0H for Token-Ring MAC frames and 04H for NETBIOS MAC frames.

Covering all LLC details here would mean repeating the IEEE 802.2 specification. The IEEE 802.2 specification contains over 100 pages detailing all primitives, operations, and field formats down to the bit level. Elements of the local network LLC procedures for code-independent data communication using the LLC PDU structure are

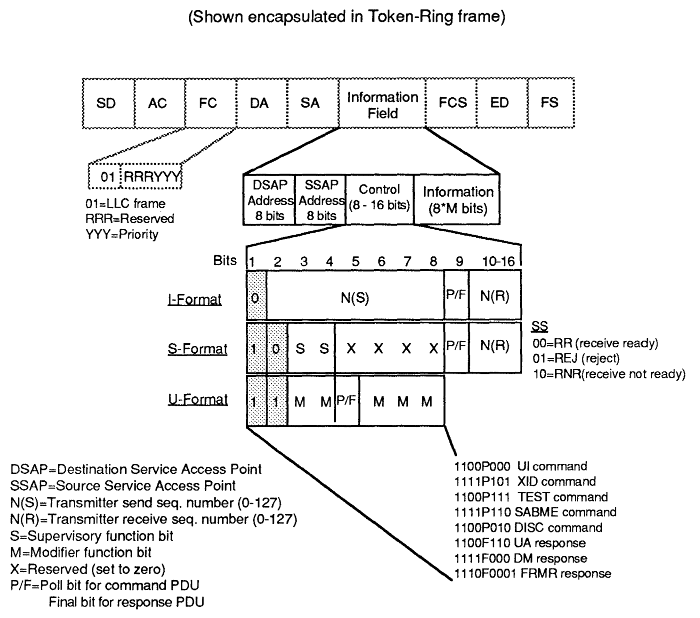

Figure 3-14: 802.2 LLC Packet Format

specified. State diagrams and transition flows are given for Type 1 (connectionless) and Type 2 (connection-oriented) operation.

LLC Type 3

LLC Type 3 operation is an IEEE 802.2 draft (802.5H -- Recommended Practice for LLC Type 3 Support) for acknowledged Type 1 operation. The following is a proposed recommended practice for the support of the 802.2 Logical Link Control Type 3 operation, prepared by the IEEE Token-Ring Rapporteur on Immediate Response. If approved, this recommended practice will be incorporated into the Token-Ring standard as Appendix B.

The appendix describes the characteristics and options which are recommended in Token-ring networks to ensure optimum support for LLC Type 3 in real time applications. Token-ring networks employing LLC Type 3 procedures typically will do so in order to support real time requirements. For the purposes of this appendix, "real time" implies an ability for any station in a network to respond to any other station in the network in less than 10 milliseconds.

Token-ring Implementation Issues: token-ring networks supporting LLC Type 3 shall have the following characteristics.

All stations shall limit maximum packet lengths to the order of 100 octets. Longer frames can be supported if the priority mechanism is employed for LLC Type 3 messages, provided these longer frames are sent at lower priority than the LLC Type 3 frames.

LLC Type 3 requests and response packets shall be 16 to 40 octets.

The maximum number of stations on a ring shall be less than 128.

Only a portion of the stations on the ring shall generate traffic at any point in time--LLC 3 or background traffic. A number less than 32 generally meets this requirement.

The length of the ring shall be kept short, less than 2 km unless early token release is employed. For networks using early token release, the ring length can be increased up to 20 km with satisfactory operation.

Station Implementation Issues: token-ring stations implementing LLC Type 3 messages shall:

- Have the ability to employ the MAC priority mechanism.
- Be able to receive, recognize, and respond to an LLC Type 3 message with minimum delay. For the maximum quantities expressed in this appendix, a value of 1 ms or less to receive a request, generate, and transmit the response is required for a successful implementation.
- Be able to use the early token release operation in long rings with high utilization.

Chapter 4

Interfaces

One of the key questions facing software developers is what token-ring interface should be written to. For the IBM PC Token-Ring adapter operating under PC DOS, there are four major interfaces to choose from: NETBIOS (Network Basic Input/Output Systems), APPC/PC, 802.2 DLC, and 802.5 MAC (direct). For OS/2, IBM offers these plus additional interfaces via the Communications Manager found in the OS/2 Extended Edition (see Chapter 6).

This chapter examines the various PC DOS interfaces. Keep in mind that the functionality of the DOS interfaces also carries over into OS/2.

PC Adapter Interfaces

All interfaces to the IBM PC Token-Ring Adapter require an adapter support interface (handler) once known as TOKREUI (Token-Ring Extended User Interface). TOKREUI was included with the adapter card on a 5-1/4" diskette as TOKREUI.COM. It is now incorporated as part of the IBM Local Area Network Support Program for DOS that was announced at the same time as Personal System/2 and also incorporated into the OS/2 EE Communications Manager.

IBM LAN Support Program

The LAN Support Program is designed to provide an interface to NETBIOS and IEEE 802.2 LLC and to enforce interface consistency for the various IBM PC LANs: Token-Ring (all PC and PS/2 adapters), PC Network Broadband (Adapter II and II/A only), and PC Network Baseband (a 2-Mbps StarLAN-like network). All models of IBM PCs are supported from the XT to AT to PS/2. Both XT and AT versions of the 3270 PC are supported except for model 5271.

The equivalent of TOKREUI (as well as NETBEUI for NETBIOS and TOKR3270 for an IBM 3270-PC version of TOKREUI) is supplied by the PC LAN Support Program. Unlike TOKREUI, which is a "terminate stay resident" program, device drivers used for the appropriate adapter/interface is loaded by DOS at boot time. DOS 3.3 or higher is required for proper operation of the device drivers.

The purpose of the adapter support interface (handler) is twofold: to supply the Token-Ring interface to canned programs such as NETBIOS and to relieve the programmer of the Token-Ring adapter complexities when designing custom applications. The handler essentially provides a direct (802.5 MAC) and logical link sublayer (802.2 LLC) of the data link layer (DLC) interface for the adapter as indicated in Figure 4-1.

There are a total of six device drivers supplied by the Support Program. One device driver called the interrupt arbitrator, is always used regardless of the adapter. It has one parameter that defines the language that load-time error messages are displayed in. In the first release of the Support Program, this parameter has no effect and error messages are displayed in English.

The other five drivers are as follows: two for Token-Ring; two for PC Network broadband or baseband; and one for NETBIOS. A program called DXMAID can be used by the user to automatically generate or modify the CONFIG.SYS file with the appropriate drivers and associated parameters for the desired adapter and optional NETBIOS support. The manner in which these device drivers fit into the Token-Ring is illustrated in Figure 4-2.

Token-Ring Driver

The two PC-DOS Token-Ring drivers are called DXMC0MOD.SYS and DXMC1MOD.SYS. The latter is used if operating the 3270 Workstation Program. The standard driver occupies 8Kb of RAM and the 3270 version occupies 14Kb. Both drivers support up to two adapters.

Adapter addresses and shared RAM locations are optional parameters for both drivers. If an adapter address is given, it overrides the universally administered address (the one burned into the adapter -- a 48-bit address that is uniquely assigned by a vendor who has obtained a range of addresses from IEEE) with a locally administered address. Locally administered addresses are handy for supporting applications

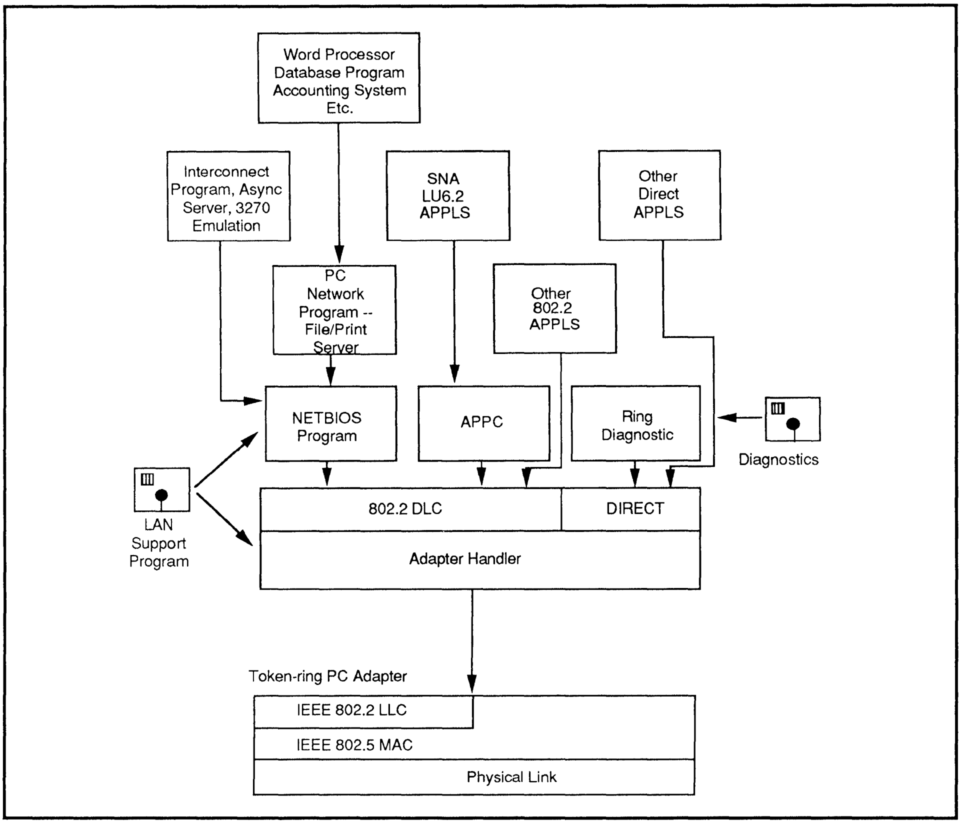

Figure 4-1: Token-Ring Interfaces

The CONFIG.SYS will contain the following information: DEVICE=DXMC0MOD.SYS na0,sr0,na1,sr1. The parameters are optional, supplied in hex notation, and have the following meaning:

na0/na1: Node address for Adapter 00/01. If entered, it must be 12 hex digits in the range '4000 0000 0000' to '4000 7FFF FFFF'. The address chosen must be unique across the entire LAN, including those that may be connected via bridges.

sr0/sr1: Shared RAM address for Adapter 00/01. If entered, it must be four hex digits specifying a segment address on an 16KB boundary (since Adapter II and Adapter/A for PS/2 contain 16KB of RAM).

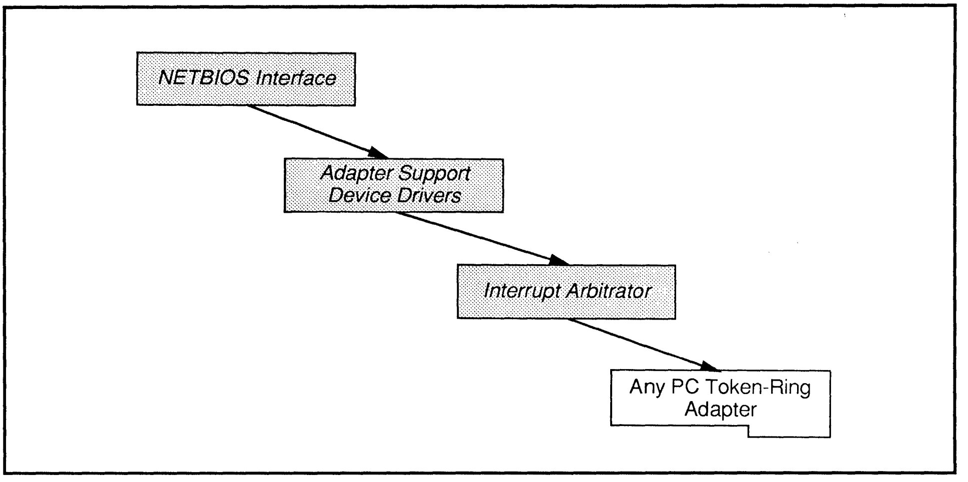

Figure 4-2: Device Drivers

sr0/sr1: Shared RAM address for Adapter 00/01. If entered, it must be four hex digits specifying a segment address on an 16KB boundary (since Adapter II and Adapter/A for PS/2 contain 16KB of RAM). If nothing is entered, a default of D800H will be used for Adapter 00 and D400H for Adapter 01.

If the device driver is loaded okay, then the handler places its entry point into interrupt INT 5CH. (Note that INT 5CH is shared with the NETBIOS interface.) The command in the command control block (CCB) is examined. If the command is less than 3, then the Token Handler takes over -- otherwise it may be for NETBIOS.

Direct Interface

The direct interface allows an application to obtain error status and logs, and to generally control the Adapter. The programmer must refer to the Token-Ring Adapter Technical Reference Manual for programming information.

Primitives (commands) are included to configure and manage the Adapter microcode and to support auxiliary commands that control buffer management, the 100 millisecond timer, and operational characteristics of the Adapter handler.

Direct commands that deal with transmit and receive buffers are:

- DIR.OPEN.ADAPTER - allocates the direct interface buffer pool

- DIR.MODIFY.OPEN.PARMS - changes direct interface buffer pool allocation
- DIR.OPEN.SAP - allocates DLC interface buffer pool for a specific service access point
- BUFFER.GET - obtain one or more buffers from a service access point (SAP) pool for later transmit use
- BUFFER.FREE - converse of BUFFER.GET

Direct commands for receiving data are:

- RECEIVE - Sets up queue to receive data for the STATION_ID of the CCB
- RECEIVE.MODIFY - Same as RECEIVE except: only non-MAC frames may be received; data is read into one SAP buffer and one user buffer; the format of the received data is different
- RECEIVE.CANCEL - Cancel receive command for SAP or link station

Direct commands for transmitting data are:

- TRANSMIT.DIR.FRAME - Transmits data for direct station -- application must prepare the frame format including addresses
- DIR.INITIALIZE and DIR.OPEN.ADAPTER commands must be issued by an application (such as NETBIOS) after the handler is loaded. DIR.INITIALIZE will reset all of the Adapter tables and buffers in both the Adapter and the handler. Adapter diagnostics are run in both the Adapter and the PC to ensure proper status of the Adapter hardware and its interface to the PC. DIR.OPEN.ADAPTER will prepare the Adapter for normal ring operation.

DLC Interface

The data link control (DLC) interface provides the IEEE 802.2 Type 1 (connectionless) and Type 2 (connection-oriented -- see Chapter 3) service interface with link station characteristics. These services include: support for node hierarchy with the station component; the SAP (Service Access Point) components and connect components; XID (eXchange IDentification) and TEST commands issued on a 'per SAP' basis; and XID and TEST responses issued by the station component (i.e., transparent to local applications).

A link station (link connection component) is a protocol machine used to manage elements of the procedure required to exchange data with a communicating link station in an adjacent node.

The SAP is an 8 bit architected code point/address through which an application is identified to the data link (DLC) software. SAPs allow multiple applications to share the same adapter (for example APPC/PC and NETBIOS), simultaneously.

The link station ID, STATION.ID, is a two byte handler local ID number of the form nnssH returned from a DLC.OPEN.STATION or a DLC.OPEN.SAP (in which an additional parameter is provided to tell the Adapter how to handle XID components). It will be used by an application in subsequent references to a particular SAP or link station. For a link station, nn00H is the STATION.ID of the owning SAP component and --ssH represents the link component user of that SAP. For a SAP, the STATION.ID is of the form nn00H.

The STATION.ID for the direct interface is 0000H. As an option, it may be split into two STATION.IDs of the form 0001H (to handle MAC frames) and 0002H (to handle non-MAC frames) for use with other (non-IEEE) data link protocols.

Available commands for controlling the DLC include:

- DLC.OPEN.SAP - Activate a SAP and reserve a number of link stations for the SAP
- DLC.CLOSE.SAP - Deactivate a SAP
- DLC.RESET - Resets a specific SAP and associated link stations or all SAPS and associated link stations
- DLC.OPEN.STATION - Allocate resources for a link station (such as number of frames transmitted before an ack has to be received)
- DLC.CLOSE.STATION - Deactivate a link station
- DLC.CONNECT.STATION - Start or complete an exchange to place both local and remote link stations in a data transfer state
- DLC.FLOW.CONTROL - Data flow control across a SAP by setting and resetting a local busy state
- DLC.MODIFY - Modify SAP parameters (such as timeout values) without closing and reestablishing the SAP link
- DLC.STATISTICS - Reads and optionally resets DLC log; the log contains information pertaining the number of frames transmitted, number of frames received, number of frames discarded, various information about the state of the link, etc.

Commands for transmitting data using DLC include:

- TRANSMIT.I.FRAME - Transmit data for link station -- application supplies only the data portion of the frame
- TRANSMIT.UI.FRAME - Transmit unnumbered information data for a SAP -- no sequence numbers are used
- TRANSMIT XID.CMD - Transmit XID command
- TRANSMIT XID.RESP.FINAL - Transmit XID with final bit on
- TRANSMIT.XID.RESP.NOT.FINAL - Transmit XID with final bit off
- TRANSMIT.TEST.CMD - Request adapter to transmit a Test command frame

The receive commands for DLC are the same as the direct interface.

Figure 4-3 helps to clarify which parts of the various frame formats must be supplied by the application for the direct and DLC interface.

The command control block (CCB) is used as the interface between the handler and the application. The application initiates a command by issuing the 5CH interrupt with the PC's 8088 or 80286 EX:BX register pair pointing to the CCB. The basic structure of the CCB is illustrated in Figure 4-4.

When a valid CCB is received, the handler will set the return code in the CCB to FFH indicating "command in process." The application must not change the CCB until the return code is set to something other than FFH by the handler.

A user appendage is an exit point to a subroutine where the handler can transfer information to the application asynchronously. Information is transferred upon completion of a command or error condition and an interrupt to the PC is generated. The appendage is entered via a CALL FAR instruction with interrupts masked off. The programmer is warned the appendage code must be reasonably fast, or else timer ticks (18.2 per second) will be lost. The appendage issues an IRET to return.

When the appendage is entered via a handler interrupt, the following state exists: register pair CS:PC is set via the CALL FAR instruction; register pair SS:SP defines the current stack (the handler uses an internal stack on all interrupt routines); all 8088/80286 registers are saved on the stack and available for use by the appendage; if the appendage was taken as a result of an interrupt, all further interrupts are

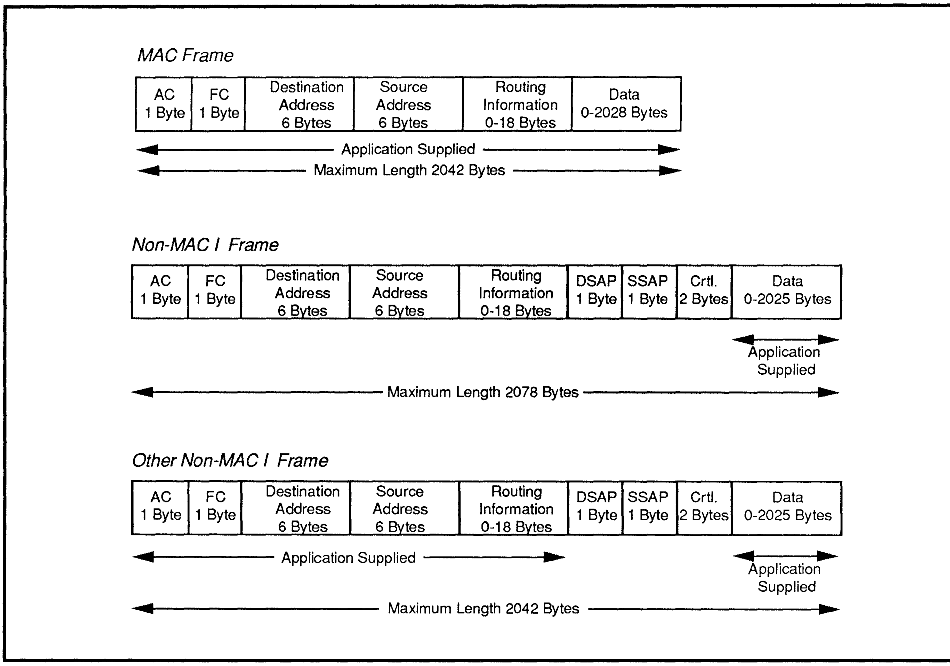

Figure 4-3: Frame Formats

Offset	Field Name	Length (Bytes)	8086 Type	Description
0	CCB_Adapter	1	DB	Adapter 0 or 1
1	CCB_Command	1	DB	Command Field
2	CCB_Retcode	1	DB	Completion Code
3	CCB_Work	1	DB	Adapter Support Interface Work Area
4	CCB_Pointer	4	DD	Queue Pointer and Adapter Support Interface Work Area
8	CCB_CMD_CPLT	4	DD	Command Completion User Appendage
12	CCB_PARM_Tab	4	--	Parameters or Pointer to CCB Parameter Table

Figure 4-4: Command Control Block (CCB)

blocked (i.e., timer tick, keyboard, etc.) until the appendage issues an IRET; register AX contains the return/status code; register CX contains the Adapter number (00 or 01) with register pair ES:BX pointing to the appropriate CCB; register pair DS:SI contains the address of the receive CCB for a RECEIVE.DATA interrupt; SI contains

USER.STAT.VALUE as defined in the DLC.OPEN.SAP CCB for a DLC.STATUS interrupt.

The following is a table of user appendages and when they are defined.

Appendage	When Defined
Command Completion	Per each CCB
Adapter Check	DIR.INIT/DIR.OPEN.ADAPTER
Ring Status Check	"
PC Error	"
Received Data	Receive Command
DLC Status Change	DLC.OPEN.SAP

The handler manages buffer pools within the PC's memory as allocated by the application. Buffer pools are associated with SAPs and are defined by the DLC.OPEN.SAP for a given SAP or by DIR.OPEN.ADAPTER for a direct station SAP. All link stations in the ring associated with the same SAP share the same buffer pool. The buffer pool format is shown in Figure 4-5. A link station will continue to receive frames while buffers are available and a RECEIVE command has been issued by the application. It is interesting to note while the RECEIVE command requires a buffer pool, it is optional for the SEND command (handy for doing a "chained" send).

After the application issues a RECEIVE command and the Adapter receives a frame, the following occurs: The Adapter will interrupt the handler; the handler will use the BUFFER.GET code to get the appropriate number of buffers from the (SAP) buffer pool; the handler will move the data from the shared RAM to the buffer(s) (if the buffer is too short, another one is automatically requested), the handler will exit via the user appendage defined by the RECEIVE command (the received data appendage), with the register pair ES:BX pointing to the first (only if the received frame fit) receive buffer. The frame is now in the buffer and the application sees it as shown in Figure 4-6. After the application "uses" the frame, it must quickly free up the buffer by issuing a FREE.BUFFER command to the handler to avoid losing frames due to lack of memory.

When an application wishes to send data, it may use its own buffer space or space provided by the buffer pool (by issuing a BUFFER.GET command). Transmit buffer format is shown in Figure 4-7. For efficiency in connection-oriented data transfer, the application

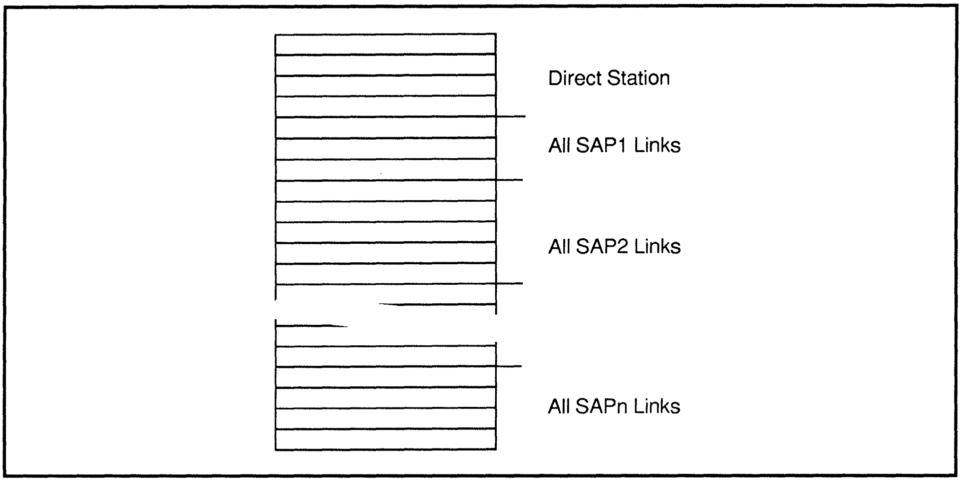

Figure 4-5: Buffer Pool

may specify the number of outstanding transmits (by setting the MAXOUT parameter) before the transmit complete interrupt is posted. If MAXOUT is not specified, the default is eight. When the transmit complete user appendage is taken, all CCBs associated with the transmit are chained together. At that time, the following state exists: the register pair ES:BX points to the first transmit CCB issued; offset four of the CCB contains the offset of the next CCB in the chain; offset six of the CCB contains the segment of the next CCB in the chain; all CCBs in the chain are marked complete and contain the same return code value; the user appendage taken is the command complete appendage of the first CCB.

To aid in understanding the operation of the handler, a number of flow diagrams (Figures 4-8 to 4-12) will be shown. The first flow diagram (Figure 4-8) shows actions taken when the Adapter is initialized for the first time. Three "programs" are involved: the application, the handler, and the Adapter microcode.

The initialization sequence takes up to 27 seconds for the first active Adapter on a token-ring, and 10 to 20 seconds for each additional activated Adapter. During this time, diagnostics are performed including: a self-test on the Adapter hardware, a loop-back test to/from the MAU, and a check to make sure a monitor station exists. If at any time diagnostics fail, the handler will try three times before reporting a failure back to the application.

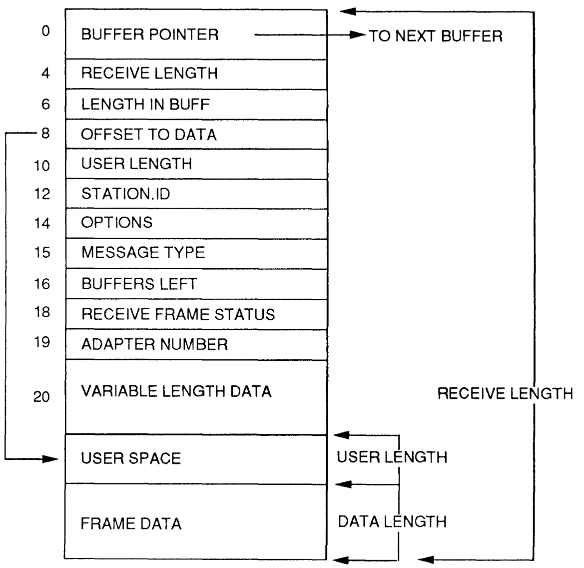

Figure 4-6: Receive Buffer

After opening a SAP via DLC.OPEN.SAP, connectionless service is available. For connection-oriented service, the DLC.OPEN.STATION command must be used. Its flow diagram is shown in Figure 4-9. One obscure acronym shown in the flow diagram, needing further explanation is SABME (Single Asynchronous Balanced Mode Extended). It is an IEEE defined procedure which basically "wakes up" the other station and says "I want to talk to you." The number of simultaneous connection-oriented services used is dependent on how the 8KB shared RAM is allocated, a practical number is 16.

Figure 4-10 is a flow diagram for transmitting data. Note that the BUFFER.GET command is optional.

Figure 4-11 is a flow diagram for receiving data. The RECEIVE command is issued first; if not the handler will discard frames. The SAP pool must also have been allocated once by the application before the RECEIVE command is issued. An interesting feature of the Adapter is if connection-oriented services are being performed, a frame will be held in shared RAM until the handler determines the SAP pool has a free buffer.

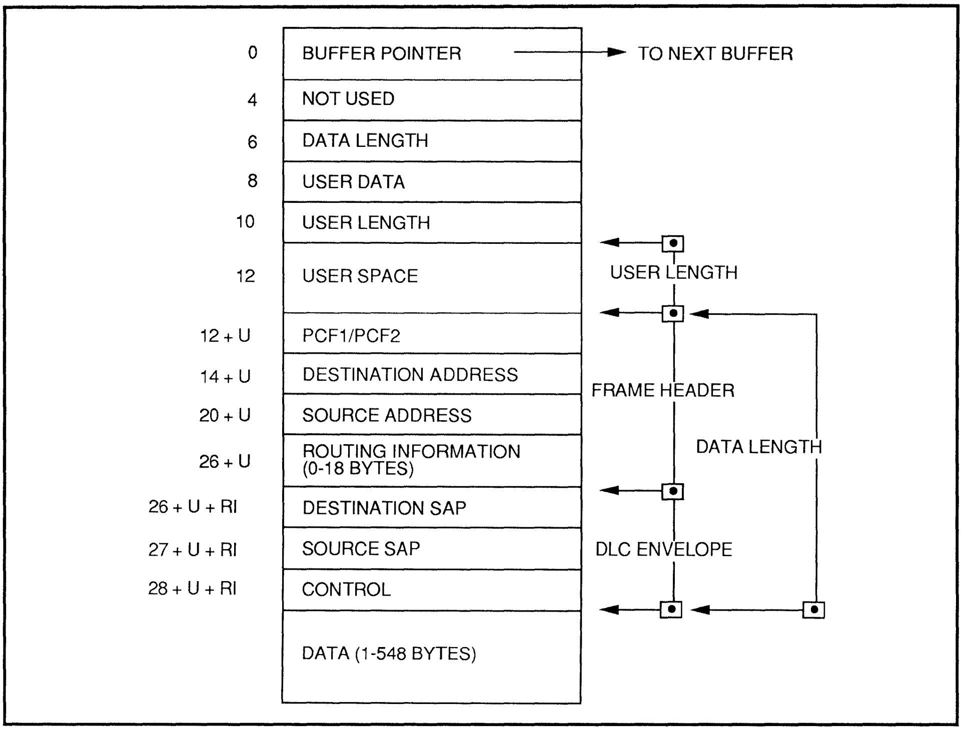

Figure 4-7: Transmit Buffer

The flow diagram shown in Figure 4-12 illustrates termination of the Adapter in which it is taken off the ring. Closing the adapter erases all tables and buffers associated with the handler. Therefore one must be very careful in a multi-tasking environment where another task may still be active. Starting up again (Figure 4-8) may require up to 27 seconds.

NETBIOS

Network Basic Input/Output System (NETBIOS) was originally developed for the IBM PC Network by Sytek (now Hughes LAN Systems). The PC Network version operates with Sytek LocalNet 20 protocols and provides an interface for various levels of protocol to the host. IBM reiterated the importance of NETBIOS by offering a NETBIOS emulator with the Token-Ring. This allows applications originally developed for the PC Network to be run on the Token-Ring. Most services provided are at the session level. Supported session services include peer-to-peer communication and naming.

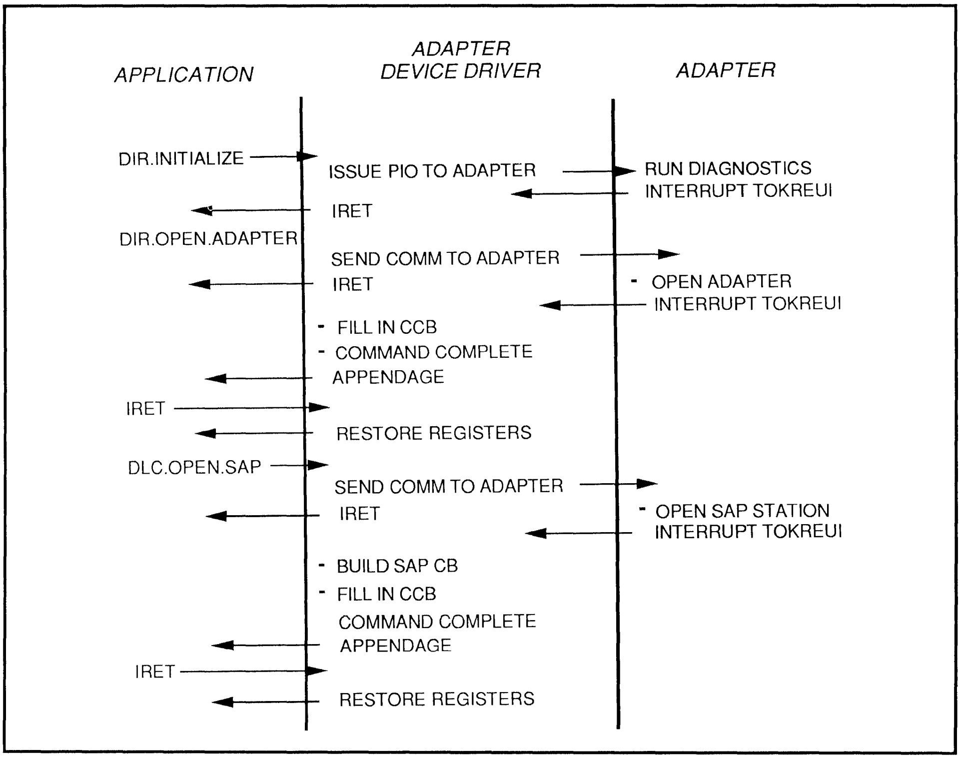

Figure 4-8: Token Handler Command Flow - Starting Up

Since NETBIOS is backed by IBM, many developers of LAN software and hardware have developed NETBIOS interfaces (also called NETBIOS "emulators").

The IBM PC Local Area Network Program (formerly the IBM PC Network Program) is an example network application that relies on NETBIOS for its operation. It implements the Server Message Block (SMB) protocol. The PC LAN Program provides the user with workstation functions (redirector, receiver, and messenger) and non-dedicated server functions (workstation and server functions). The PC LAN Program is covered in more detail in Chapter 5.

It should be noted that the NETBIOS interface option available for the Token-Ring is an "emulation" of the NETBIOS contained within the PC Network adapter card. Therefore, the actual protocols used within the various layers differ between the Token-Ring and the PC Network but what the user or programmer will see are identical in-

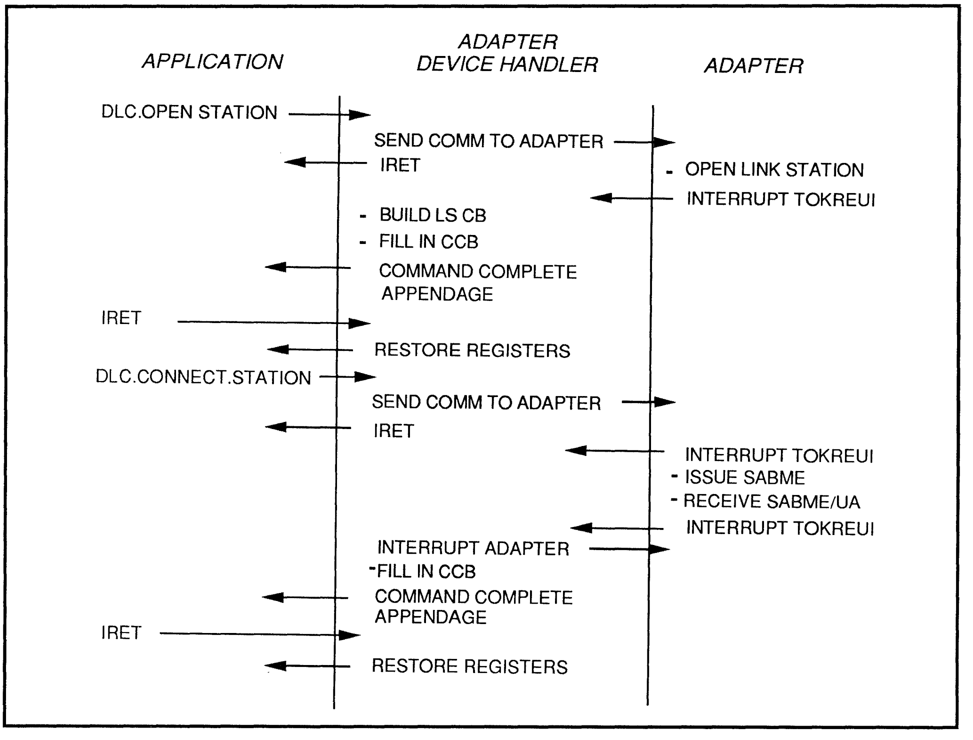

Figure 4-9: Token Handler Flow - Link Connection

terfaces and operation (except for response times, which are much faster in the Token-Ring).

Figure 4-13 shows how NETBIOS is implemented in the two networks. Note that on the Token-Ring, the host processor must execute the protocols, whereas on the IBM PC Network, an on-board 80188 processor does the protocol processing. Interestingly, NETBIOS tests performed by IBM on both networks have shown the Token-Ring implementation to operate by more than a factor of two (the raw data rate gain from PC Network (2 Mbps) to Token-Ring (4-Mbps)) better than PC network. This is partly due to overhead in the PC Network adapter design, and the way in which NETBIOS protocols were programmed. With the Token-Ring NETBIOS implementation, the transport and network layer overhead is eliminated since these layers are essentially non-existent. The Token-Ring version uses the 802.2 LLC connection-oriented services (Type 2) for all but datagrams which rely on 802.2 connectionless services (Type 1).

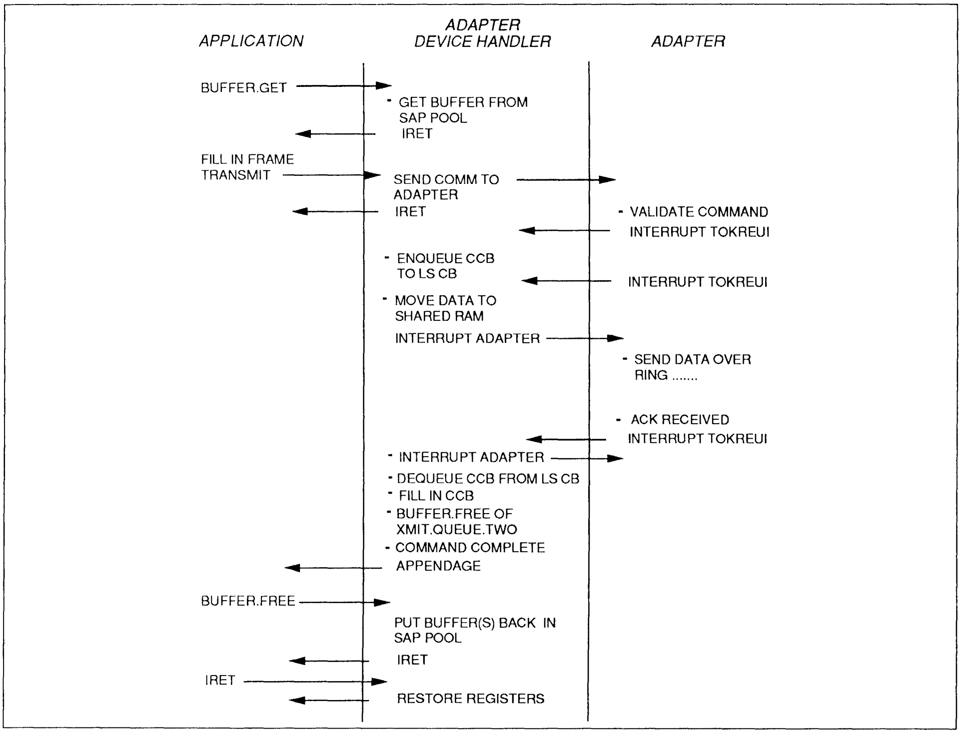

Figure 4-10: Token Handler Cmd Flow - Transmit Data

The host communicates to NETBIOS via the Network Control Block (NCB). (The NCB is also called the Message Control Block or MCB in the IBM Token-Ring Network PC Adapter Technical Reference manual.) The format and operation of the NCB is detailed later in this Chapter. Once the NCB is set up by the host, it interrupts NETBIOS for service. NETBIOS then takes over and invokes the necessary protocols to perform the service requested by the host (this service may not require protocols, i.e., a request to perform local diagnostics or to obtain the adapter's address).

The NETBIOS session layer interface provides host access to several protocols. Session management provides support for user sessions between nodes allowing users to establish connection to a named process and is responsible for determining the named process address. Once the destination node is determined, the initiating application can communicate with the destination node to provide session level ser-

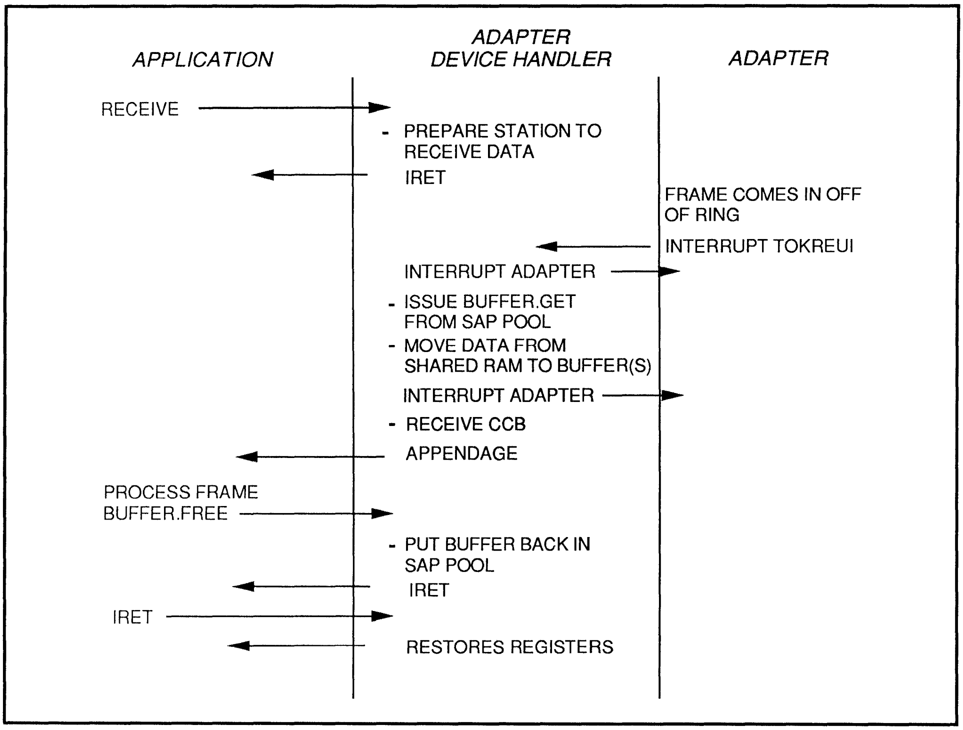

Figure 4-11: Token Handler Command Flow - Recieve Data

vices. In conjunction with naming, a datagram service is available for sending datagrams between two names (nodes).

NETBIOS name management provides binding of alias names and network addresses within the entire local network providing all name management services, including the translation of remote names to a network address. This mechanism of NETBIOS is one reason why it takes so long to initially become part of a NETBIOS network -- the node will broadcast (a number of times to ensure reception by all other stations on the network) its name(s) to other stations -- this ensures that the node has a unique name. Broadcasting also occurs to establish a session connection with another name.

Once supplied as the NETBIOS Extended User Interface (NETBEUI), NETBIOS is now configured into the IBM PC LAN Support Program as a device driver (as discussed at at the beginning of this chapter). Applications (i.e., the PC Network Program) written to NETBIOS for PC Network will work "as is" with the Token-Ring

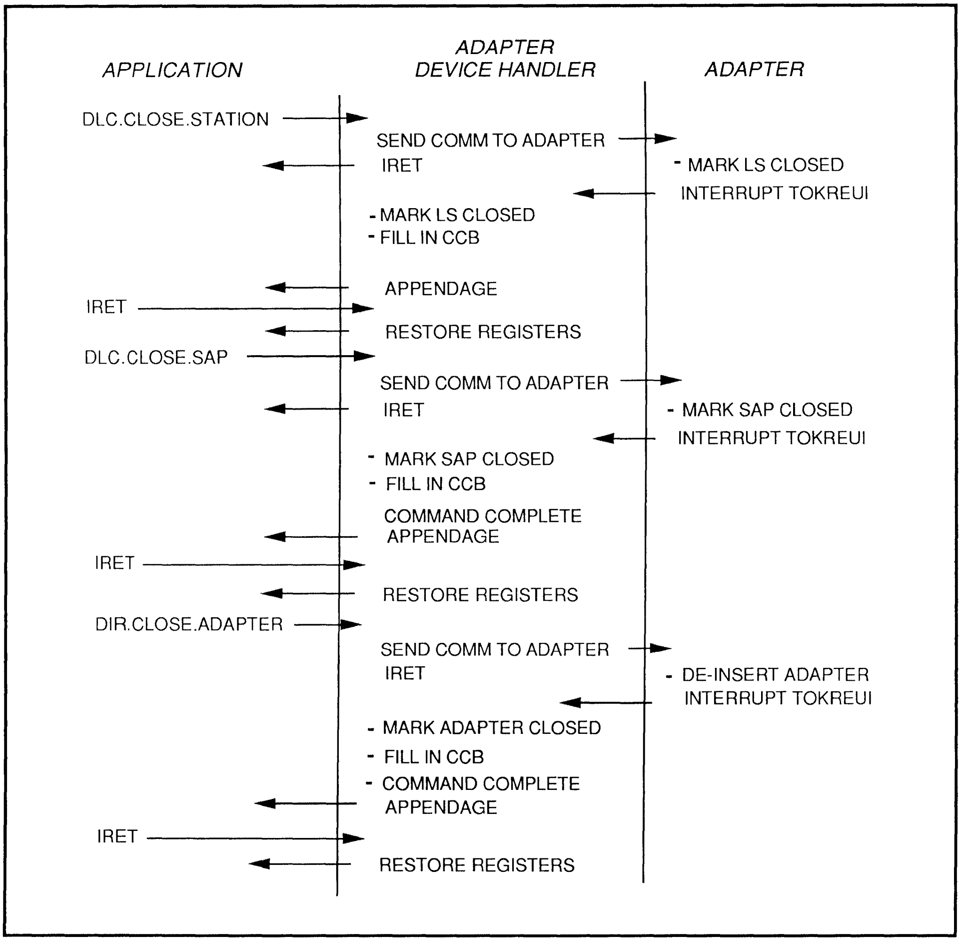

Figure 4-12: Token Handler Command Flow - Shut Down

emulator. The device driver occupies 23Kb of RAM in its default configuration.

Driver

With the PC LAN Support Program, NETBIOS has more parameters than with earlier releases. Support for earlier releases of NETBIOS is provided such that all previous parameters may be used (as a migration aid).

One of the new options provides support for high-speed asynchronous adapters -- parameters may be supplied called ENABLE that essentially allocates less time to NETBIOS and more

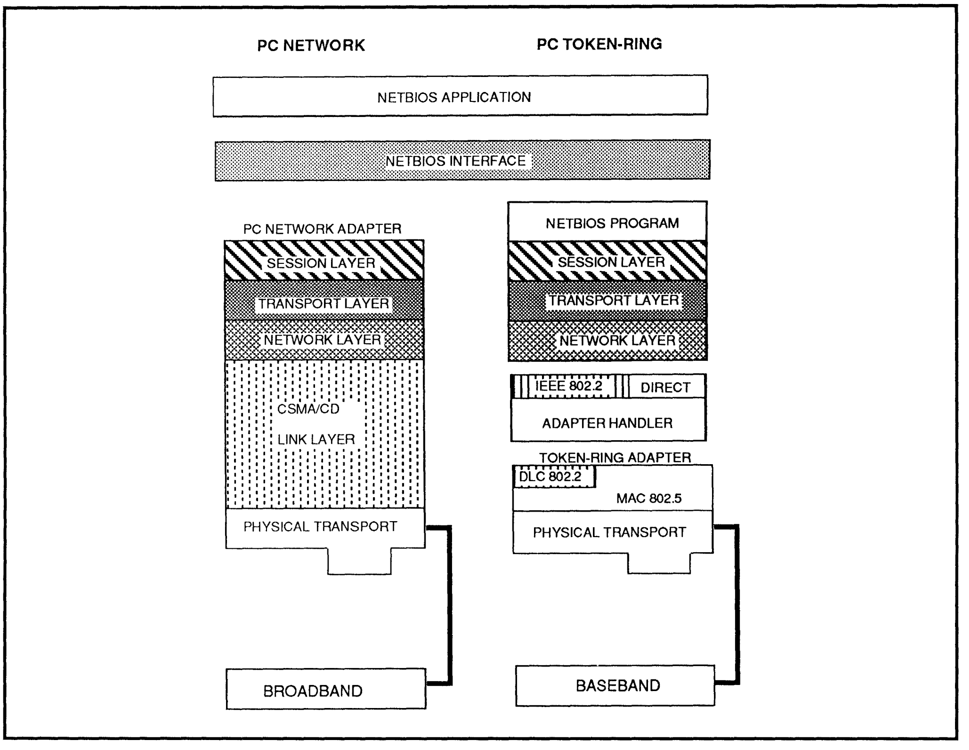

Figure 4-13: NETBIOS Implementation

time for asynchronous adapter operation -- thus slowing down NETBIOS performance.

The table in Figure 4-14 is a summary of all of the available parameters. The parameters are supplied via DEVICE= DXMT0MOD.SYS in the CONFIG.SYS file and are not position dependent (like the old NETBEUI). An example would be DEVICE=DXMT0MOD.SYS ST=50 N=40 S=30 to support 50 stations, 40 names, and 30 sessions.

Programming

Similar in manner to an application invoking the direct and DLC services, an application requiring NETBIOS services will set up a Network Control Block (NCB -- also known as the Message Control Block or MCB) and issue a 5CH interrupt. The NCB structure and the general meaning of each field is illustrated in Figure 4-15.

Keyword	ABBR	Valid Values	Minimum	Default
Stations	ST	0 - 254 *	1	6
Sessions	S	0 - 254	1	6
Commands	C	0 - 255	1	12
Names	N	0 - 254	2	17
Open.On.Load	O	Yes/No	-	Yes
Datagram.Max	DG	Yes/No	-	No
Close.On.Reset	CR	Yes/No	-	No
DHB.Size	DS	200 - 9999 *	200	**
DHB.Number	DN	0 - 9 *	-	**
Receive.Buffer.Size	R	0 - 9999 *	***	**
Transmit Timeout	TT	0 - 20	0	2
Transmit.Count	TC	0 - 1-	0	3
DLC.Maxout	MO	0 - 9	0	2
DLC.Maxin	MI	0 - 9	0	1
Ring.Access	RA	0 - 7	0	0
Extra.Saps	ES	0 - 99 *	0	0
Extra.Stations	EST	0 - 99 *	0	0

* Making this value too large will cause a failed adapter open.

** If default value is used, the NETBIOS Driver sets the values of these items based on adapter resources.

*** Minimum is set by adapter on open, not NETBIOS.

Figure 4-14: NETBIOS Device Driver Parameters

When an application issues commands to NETBIOS, it may choose to wait for completion or be interrupted upon completion. A listing of the possible commands follows.

Command	Function
General Commands	
RESET	Resets local adapter status and clears name and session tables.
CANCEL	Requests cancellation of a pending command whose NCB can be found at NCB_BUFFER@
STATUS	Gives status information for local or remote Adapter. Status includes: software version, traffic and error statistics, and local name table.
UNLINK	Used with remote program load (RPL) to unlink session with IBMNETBOOT.
ADD NAME	Adds a unique, 16-character name to name table.
Name Support Commands	
ADD GROUP NAME	Adds a group name to name table.

Field Name	Length (in bytes) and Meaning	
NCB_COMMAND	1	- NCB COMMAND FIELD
NCB_RETCODE	1	- NCB RETURN CODE FIELD
NCB_LSN	1	- NCB LOCAL SESSION NUMBER FIELD
NCB_NUM	1	- NCB NUMBER OF YOUR NAME
NCB_BUFFER@	4	- NCB POINTER TO MESSAGE BUFFER ADDRESS (OFFSET:SEGMENT)
NCB_LENGTH	2	- NCB BUFFER LENGTH (IN BYTES)
NCB_CALLNAME	16	- NCB NAME ON LOCAL OR REMOTE ADAPTER - FOR CHAIN SEND THE FIRST 2 BYTES INDICATES LENGTH OF SECOND BUFFER THE NEXT 4 BYTES INDICATES THE BUFFER ADDRESS SECOND
NCB_NAME	16	- NCB NAME ON LOCAL ADAPTER
NCB_RTO	1	- NCB RECEIVE TIMEOUT VALUE
NCB_STO	1	- NCB SEND TIMEOUT VALUE
NCB_POST@	4	- NCB POINTER TO POST ROUTINE (OFFSET:SEGMENT)
NCB_LANA_NUM	1	- NCB ADAPTER NUMBER; 00H FOR FIRST ADAPTER, 01H FOR SECOND ADAPTER
NCB_CMD_ CPLT	1	- NCB COMMAND STATUS FIELD
NCB_RESERVE	14	- NCB RESERVED AREA

Figure 4-15: Network Control Block (NCB)

DELETE NAME	Deletes name from name table.
FIND NAME	Finds the location on the network of the 16-character name.

Session Support Commands

CALL	Opens a session with another name specified by the NCB_CALLNAME and filed using the local field specified by the supplied NCB_NAME. The called name must have issued a LISTEN command.
LISTEN	Enables a session to be established with the name specified in the NCB_CALLNAME field.
HANG UP	Closes the session with another name.
SEND	Sends data by the session number indicated in the local session number (LSN). The data is taken from the buffer indicated. A maximum of 65KB may be sent.

CHAIN SEND	Like SEND, except data is taken from the buffers for the indicated number of bytes. Two buffers can be chained together.
RECEIVE	Receive data from a specified session. Time-out values can be specified.
RECEIVE ANY	Receive data from any station with whom you have a session.
SESSION STATUS	Obtain status of all active sessions for your name.

Datagram Support

SEND DATAGRAM	Send a datagram to a unique name or DATA (return) group name at a local or remote node.
SEND BROADCAST DATAGRAM	Send message to everyone who has a RECEIVE BROADCAST DATAGRAM outstanding.
RECEIVE DATAGRAM	Receive a datagram message from any name or anyone on the network directed to this station.
RECEIVE BROADCAST DATAGRAM	Receive a datagram from anyone show issues a SEND BROADCAST DATAGRAM command.

Miscellaneous

TRACE	Activates a trace of NCBs issued to NETBIOS and some of the Token-Ring handler CCBs issued by NETBIOS itself, include transmits and receives.
UNLINK	Provided for NETBIOS compatibility -- the Token-Ring implementation treats this as a "no-operation."

As noted in Chapter 3, NETBIOS on the original PC Network (broadband) adapter does not implement the standard 802.2 LLC or MAC. Therefore, on the Token-Ring, NETBIOS has been assigned an architected functional address of 00000080H to satisfy 802.2 requirements. With the NETBIOS program operational, all adapters with the functional address set will receive all frames destined for that address. The architected SAP value is F0H. Frames destined for DLC

SAP F0H will be routed to the NETBIOS program whether received through functional address or specific node address detection.

The following are considerations of the NETBIOS Token-Ring. Initializing the Adapter handler can be done explicitly by an application in which an established shared RAM address is used and the error appendages are defined either by the application or implicitly by the NETBIOS program when a RESET or the first NCB is encountered. In this case, shared RAM addresses D8000H/D4000H for adapters 00/01 will be used and NETBIOS will define error appendages.

OPEN CCB is an optional NETBIOS call used to define a set of NETBIOS program specific parameters. OPEN CCB can be done explicitly by an application, and must be issued before the first NCB and after NETBIOS is loaded. OPEN CCB can be done explicitly by a RESET or first NCB.

A typical initialize sequence would be as follows: the NETBIOS device driver issues to the Token-Ring handler DIR.INITIALIZE, DIR.OPEN.ADAPTER, DIR.STATUS, DLC.OPEN.SAP (with SAP set to F0H), DIR.SET.FUNCTIONAL.ADDRESS, DLC. MODIFY, and SET.TIMER.

The following is the sequence of NETBIOS events occurring when an application issues a NETBIOS command (via the NCB) to establish a session: the NETBIOS device driver issues to the Token-Ring handler DIR.SET.TIMER (for name recognized response), issues DIR.TRANSMIT.UI (broadcast NAME.QUERY), returns immediate return code if no-wait NCB, RECEIVE data response (name recognized), issues DIR.CANCEL.TIMER, issues DIR.FREE. BUFFER, issues DLC.OPEN.STATION (establish link station), issues DLC.CONNECT_STATION (connects the nodes), issues DIR.TRANSMIT.FRAME (send session initialize), RECEIVE data response (session confirmed), and returns final NCB return code.

There are a number of subtle differences in the Original PC Network (broadband) NETBIOS vs. the PC LAN Support Program emulation. The Support Program implementation contains a few enhancements which should not be used if compatibility with the PC Network is desired. The key differences are detailed here.

Support Program NETBIOS	Original PC Network (broadband) NETBIOS
254 links per Adapter	16
254 sessions per Adapter	32

254 names	17
255 outstanding commands	32

Both a PC Network adapter (either broadband or baseband) and Token-Ring Adapter may coexist in the same PC. If a PC Network adapter is present and operational, all NCB commands issued for that Adapter number will be routed to it. If a PC Network Adapter is not present, then all NCB commands issued for that Adapter number will be routed to the Token-Ring NETBIOS program. The user can jumper select which adapter is the primary (00) adapter and which adapter is the secondary (01).

APPC

Advanced Program-to-Program Communication (APPC) -- also called LU6.2, which is the technical term for it -- plays a crucial role in Systems Network Architecture (SNA). LU6.2 is also the protocol of choice for developing peer-to-peer applications in an IBM SAA environment. SNA is a high-function network architecture that supports distributed data processing, communications, and network management. SNA has led an evolutionary existence since 1974 and provides a base for future communications growth within SAA. The relationship of LU6.2 to SNA, as well as the reference model for Open Systems Interconnection (OSI), is illustrated in Figure 4-16.

The Logical Unit

The Logical Unit (LU) is the intermediary between the transport network and user devices, and application programs using or attached to the SNA network. Figure 4-17 shows the position of the LU in the layered structure of SNA.

The LU is concerned primarily with session protocols for paired end users. LUs serve as the attachment points for the end destinations of data (application programs, databases, and devices) and provide the end-to-end session protocols in support of communication between these network resources. While the transport network provides global flow control for links and intermediate nodes, the LU provides end-to-end flow control so that, for example, an application program does not send data to a printer faster than the printer can handle it. Other functions include session cryptography, name-to-address translation (using distributed directory services), and blocking and subdividing message units for network efficiency.

LU6.2

LU6.2 is supported by Physical Unit 2.1, although APPC can also function with PU2.0. In the past, intelligent devices (such as the PC) on an SNA network have been represented by PU2.0. This means that the PC merely emulated a cluster controller or a terminal, implying a master-slave relationship with the host. PU2.1 allows true peer-

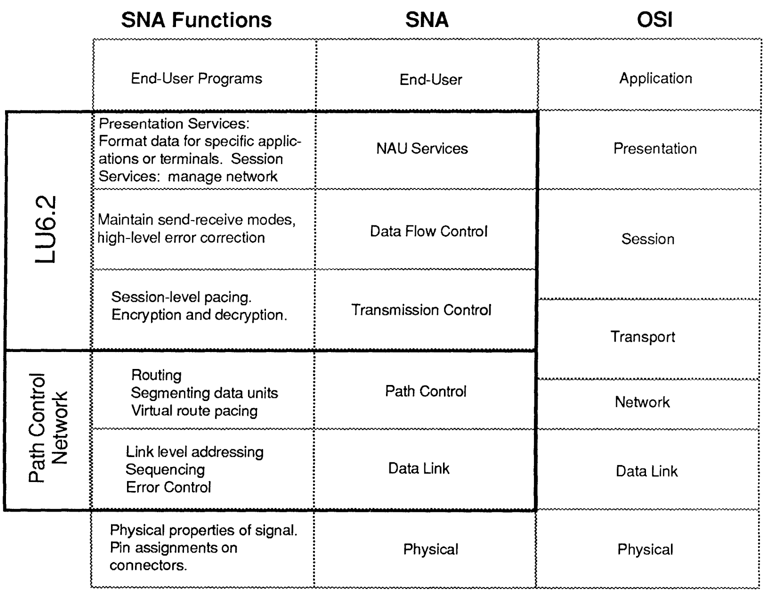

Figure 4-16: Relationship of LU6.2A to SNA and OSI

level communication in which peers can set up and break down a communications path with no host intervention. This is obviously one of the requirements of APPC.

What are some system requirements for LU6.2? First of all, it is a single architecture for all families of computers, from the Personal System/2 up to the 3090 for program-to-program communication; it allows any-to-any communication. LU6.2 can be used over any SAA transport including X.25, SDLC, and of course, the Token-Ring. The key to LU6.2 is its support for the applications that require program-to-program communication: this includes distributed data processing, file transfer, document distribution, office communications, remote database access, and network management.

IBM designed LU6.2 to be functionally complete, to use network resources efficiently, to have the ability to use it from a high-level

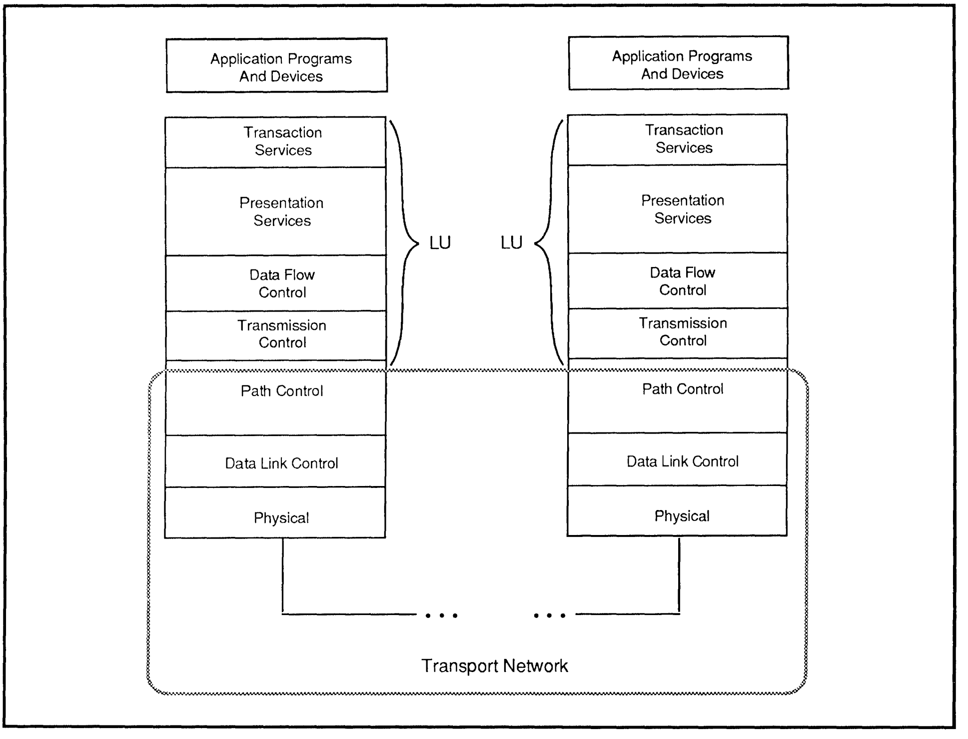

Figure 4-17: The Role of the LU

language, to insulate the high-level language programmer from underlying formats and protocols, and to offer manageability in terms of low entry costs for small systems. In other words, the size of the system executing LU6.2 is manageable by using the LU6.2 base functions, as well as incorporating options for large systems. LU6.2 has strong subset control, as well as migration features from LU6.0 and LU6.1.

LU6.2, then, is an architected protocol boundary that consists of verbs and states. Verbs represent the LU6.2 functions available to programs, while semantics define the function. LU6.2 was originally defined with semantics; IBM later added syntax so that the programming interface was consistent from machine to machine. The states of LU6.2 represent conditions at the protocol boundary and determine which verbs the program may issue. The three functional interfaces consist of sets of verbs to implement basic conversation, MAP conversation, and operator control.

The key concepts of LU6.2 warrant additional discussion: names, sessions, conversations, verbs, bases and towers, and architected protocol boundary.

LU6.2 allows transaction programs to refer to its resources, such as other transaction programs and LUs and shared communication facilities, by names. The programs need not be concerned with implementation and configuration details, such as the actual network addresses or transport characteristics. When one transaction program (TP) invokes another, the invoking TP identifies the partner TP by a transaction program name, identifies the partner LU by an LU name, and identifies the desired set of session characteristics by node name.

Sessions are used to connect logical units. Sessions provide relatively long-lived connections between LUs; a session can be used by a succession of conversations. Sessions are activated by LU pairs as a result of operator commands and transaction program requests for conversations.

A mode is a set of characteristics that may be associated with a session. These characteristics typically correspond to different requirements for cost, performance, and so forth. Modes are defined by the control operator as a selection of path-control-network facilities and LU session-processing parameters.

One characteristic of mode is class-of-service. The path control network can offer different classes of service that correspond to particular physical links and routes and particular transport characteristics such as path security, transmission priority, and bandwidth. Other characteristics of mode include operator-selected processing parameters such as message-unit sizes and the number of message units sent between acknowledgements (pacing window sizes).

Conversations which connect distributed transaction programs exist on top of these sessions (multiple conversations can share a single session). Multiple sessions between logical units are illustrated in Figure 4-18, as well as three conversations that exist on three of the sessions.

There are two types of conversations: basic (used by service transaction programs that offer services to transaction programs) and mapped (used by application transaction programs themselves). Also, there are two different programming interfaces into the LU. Basic conversations allow TPs to exchange records containing a two-byte-length prefix. Mapped conversations allow TPs to exchange arbitrary data records in any format set by the programmer.

The basic conversation interface is more flexible and offers more features than the mapped conversation interface, but it is also more difficult to use. For this reason, end-user programs typically use the mapped conversation interface, in which the LU automatically ensures conformance to certain formats and protocols. Two additional functions available in the mapped interface are data stream handling and data mapping. When mapped conversations are used, the LU puts all data into a standard data stream format called the General Data Stream (GDS). Mapping allows transaction programs to select transformations called "maps" to be performed on data at the sending and receiving nodes (for example, transforming ASCII to EBCDIC).

It is important to understand that application programs use only the

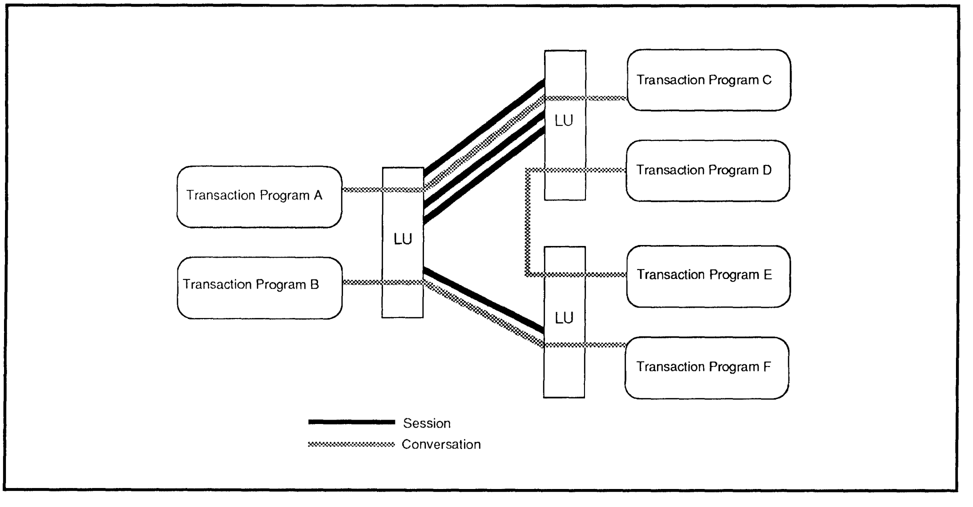

Figure 4-18: Conversations on Sessions

conversation protocols; the session protocols themselves used for LU6.2 are not visible to the program.

An APPC program is essentially a list of verbs to be executed in a set order by the LU. After the transaction program issues each verb, its processing is suspended while the LU executes the verb. There are two main kinds of APPC verbs, each with three subcategories: conversation verbs with mapped, basic, or type-independent conversations; and control operator verbs (that do not transfer data) for changing the number of sessions, controlling the session, and defining the LU.

LU6.2 is organized into subsets of functions. There are two base function sets which all open API implementations of LU6.2 must support, and there are the option function sets, or "towers." The base function sets are basic conversations and mapped conversations; the towers include sync point (for transaction backout and recovery), program initialization parameters (PIPs), transaction and session security (using IDs and passwords), and performance options (such as accelerating or eliminating acknowledgements).

The LU6.2 base is sufficient for general communications. (In fact, IBM claims that approximately 80% of all applications can be written using only the LU6.2 base.) The optional towers which sit on top of the LU6.2 base are independent and hierarchically related, and provide additional functions.

An important set of IBM architectures and software products which currently use APPC are those associated with DISOSS (Distributed Office Support Systems) -- and this includes SNA Distribution Services (SNADS), an architecture allowing store-and-forward document distribution between mainframes operating DISOSS. Machines submit documents to DISOSS using DIA (Document Interchange Architecture), which defines formats for documents. The IBM "Personal Services" series uses DIA and DISOSS facilities and is available for PC/Personal System, System/36, System/370, 43XX, and 30XX.

Products in which IBM has chosen to provide closed LU6.2 implementations include the ScanMaster 1, DisplayWriter, the 5520 to 5550, the 3820 page printer, the 8100 DPPX, and BusinessView (a PC product available only in Europe that allows access to remote data).

APPC/PC

Advanced Program-to-Program Communication for the IBM Personal Computer (APPC/PC) is a licensed program that supports the SNA application programming interface (logical unit (LU) Type 6.2, physical unit (PU) type 2.1) and allows program-to-program communication over an IBM Token-Ring Network and synchronous data link control (SDLC) communication links. As a common protocol for communication, APPC can provide improved connectivity among distributed transaction programs.

APPC/PC supports both the IBM Token-Ring Network PC Adapter and SDLC adapters. It provides a common program-to-program protocol that allows multiple conversations between applications running in an IBM Personal Computer and a System/370 (CICS/OS/VS Version 1, Release 7), System/36, System/38, AS/400, Series/1 (Realtime Programming System Version 7.1), or another IBM PC. APPC/PC does not provide direct connectivity between sessions on the IBM Token-Ring Network link and sessions on the SDLC link, but it does provide the application programming interface to allow a user-written program to communicate between sessions on the two links. It conforms to SNA LU Type 6.2. The support is based on SNA LU Type 6.2 sessions. APPC/PC supports PU2.1 architecture and provides a peer-to-peer relationship between the IBM PC and other PU2.1 nodes, such as the System/36, System/38, and Series/1. APPC/PC attaches to a System/370 as a PU2.0 node.

APPC/PC can be used by IBM PCs to help satisfy the requirements to attach to other SNA products using the SNA LU6.2 architecture. It provides communication services for applications in much the same way that IBM PC DOS provides disk input/output and file management services.

The program has been designed to have an open API. APPC/PC appears as a set of communication services provided by the operating system. It is loaded into the PC and remains resident, much like NETBIOS. An application accesses APPC services through PC DOS interrupt 68H. THE SAP assigned to APPC is 04H (F0H is assigned for NETBIOS). If NETBIOS is to be run concurrently, then NETBIOS must be loaded first, specifying an additional non-NETBIOS SAP value of 04H. APPC/PC will consume 160Kb of main memory in the PC. An additional 21Kb is required for menus, up to 5.5Kb for each additional LU, and 2.5Kb is needed for each additional concurrent conversation/session.

It supports a program interface through a set of verbs that allows application transaction programs to communicate at a conversational level with no session awareness. Application programs can engage in two types of conversation: mapped conversation -- intended for use in communications between user-written programs; and basic conversation -- intended to provide a lower level interface, such as that required by logical unit service programs.

APPC/PC security support is provided to the IBM PC application program at both the session and conversation levels.

Network management data can be created by APPC/PC or by the user application program. The data can be sent to the network problem determination application.

An IBM PC application can use both this program and the LAN Support Program NETBIOS for concurrent access to the Token-Ring Network. Only one Token-Ring Personal Computer adapter is required.

Connectivity supported by the program includes: IBM PC to IBM PC, using an IBM Token-Ring Network communications link; IBM PC to IBM PC, using an SDLC communications link; and IBM PC to System/370, System/36, System/38, or Series/1, using an SDLC communications link.

To IBM, LU6.2 is the technology, a base, and a strategy for future program-to-program communication. APPC provides a standard program-to-program communication interface (independent of operating systems) hardware, programming languages, data formats, communication protocols, and network configurations.

NETBIOS or APPC/PC?

Like most higher-level protocols, NETBIOS is hardware-independent, which makes it useful across a variety of systems. The future of NETBIOS for heterogenous systems is uncertain, however. IBM introduced APPC/PC for the Token-Ring, which has a much broader appeal over the entire line of IBM computers and communications equipment. APPC is of great importance to developers of applications for IBM computers, since IBM offers it for every major computer line it sells and is actively promoting it as an "open" interface. It is also a strategic component of IBM's Systems Application Architecture (SAA); NETBIOS is not. The communication manager component of IBM's OS/2 Extended Edition for the Personal System/2, includes not only NETBIOS, but APPC/PC as well.

NETBIOS does have the advantage of having been established in hundreds of PC LANs, while APPC/PC is still "catching up". Another edge NETBIOS has over APPC is that APPC contains complex SNA protocols which consume much of the system resources (RAM + CPU). One wants to minimize this overhead in a personal computer. This may become less of an issue however, as the more powerful PS/2 and 80386 PCs are in widespread use.

Developing applications to NETBIOS on the Token-Ring requires an interface with PC-style servers and gateways (see Chapter 5). However, it also means those applications probably won't port to other larger systems which IBM supports on the Token-Ring such as System/36 and the 9370. APPC is a safer bet for developing long term, distributed processing applications, while NETBIOS is certainly a safe bet of PC LAN only applications. This could change if IBM develops a NETBIOS emulator for non-PC computers, which appears unlikely.

According to IBM, the destiny of APPC is to become "one architecture for general purpose program-to-program communication,"and designed for "any-to-any" connectivity. Furthermore, the design of APPC is such that it has a low entry cost (ala APPC/PC) and is to be highly functional with strong subset control. APPC is IBM's foundation for developing distributed applications.

The bottom line is this: use NETBIOS to support traditional PC applications and APPC/PC for PC-to-mainframe and peer-to-peer applications; or NETBIOS for PC only LANs and APPC/PC for mixed environments that need to support IBM host applications such as those that fall under the IBM SAA umbrella.

Chapter 5

PC-DOS Services

This chapter discusses selected services offered by IBM for its Token-Ring that operate with PC-DOS.

PC LAN Program

The IBM PC Local Area Network Program (formerly called the IBM PC Network Program) was originally designed to operate with DOS 3.1 and the IBM PC Network (broadband). NETBIOS support available via the PC LAN Support Program, along with DOS 3.3 or higher, is required to operate the PC LAN Program on the Token-Ring.

The PC LAN Program consists of a single large executable file which can be initiated one of four ways: the user executes a NET START command and then specifies 1) redirector, 2) receiver, 3) messenger, or 4) server. The first three (redirector, receiver, and messenger) are for workstations, and the last one (server) is a non-dedicated file/print server implementation which runs as a background task in a workstation. Figure 5-1 shows an example implementation with one PC configured as a file server and another as a print server.

Using the redirector is the most basic way to get onto the local network. It intercepts the workstation's printer and disk I/O to send to a server; users can also send messages to other machines. The receiver, messenger and server perform the same as the redirector with the addition of: the receiver receives and logs messages to any device or file; the messenger allows a user to transfer files; and the server allows hard disks and printers to be shared.

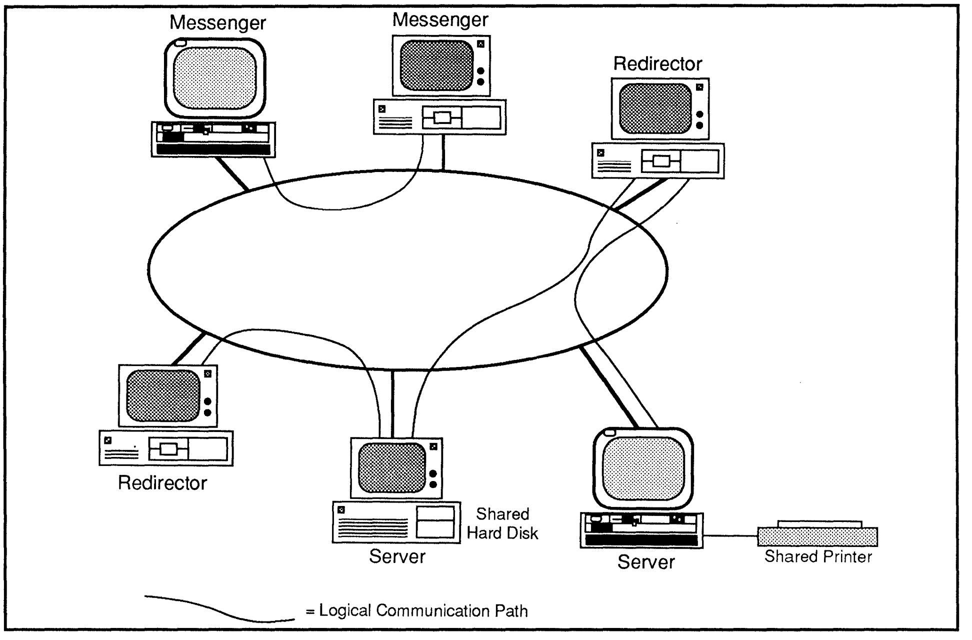

Figure 5-1: PC LAN Program Servers

The PC LAN program is an application that requires NETBIOS for its operation. Figure 5-2 presents a technical overview of DOS 3.X and NETBIOS operation. It shows how the DOS service interrupt (21H) is intercepted and if necessary, how the service request is redirected over the network via the NET program to a server to perform an operation. An application can also bypass the NET program by issuing its own NETBIOS commands via interrupt 5CH, as discussed in Chapter 4.

The original version of the PC LAN program has suffered heavy criticism -- primarily its poor throughput and lack of administrative features. The performance problem relate to the fact that unlike Novell's NetWare, PC LAN Program relies on DOS for its file and printer operation. As DOS continues to improve and the migration is made to Operating System/2, the operating system problems related to PC LAN Program will become less of an issue. DOS 3.3, for example, has added additional file allocation table (FAT) caching (the FASTOPEN command) and PC LAN Program Version 1.2 added disk caching.

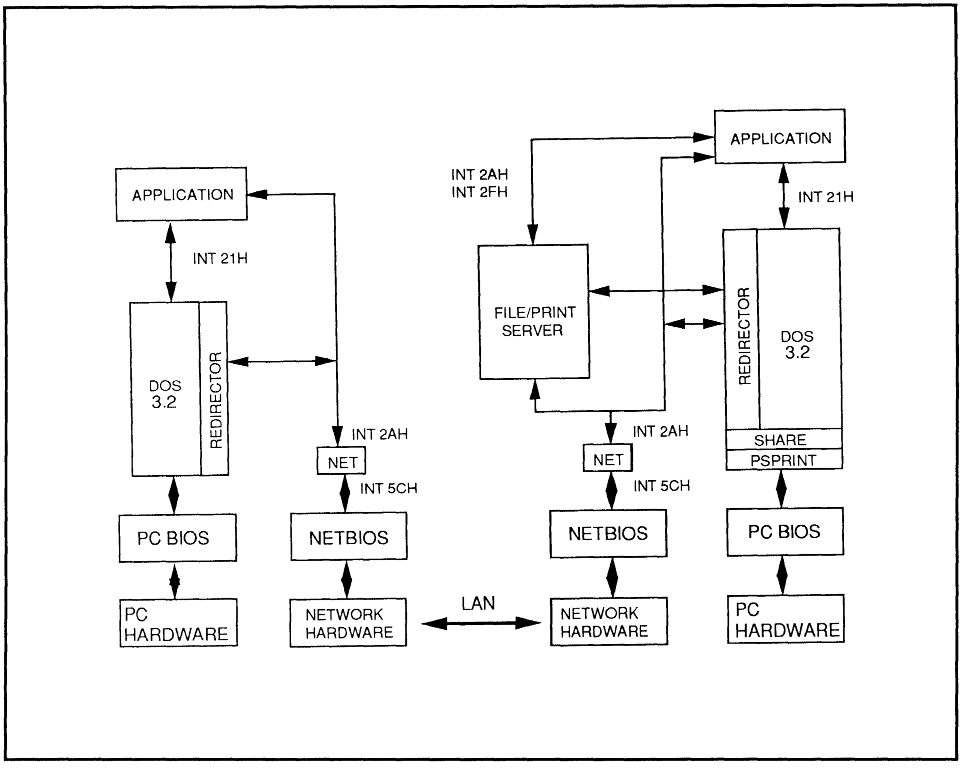

Figure 5-2: DOS/NETBIOS

PC LAN Program Version 1.3 has added the following features to address the administration problem: password controlled access (login) to a server; central resource definition and control including setting the time/date on a master server to synchronize dates and times in all workstations and servers; printer management; define of users and modify privileges; manage application selector menus for individual users; administrator access to resources from any workstation; support for remote program and operating system load (i.e., support for diskless workstations); ability to view logged on users; and finally, remote workstation printer selection, queuing, and status. Version 1.3 has also made small improvements in performance.

Asynchronous Communications Server

A service, introduced by IBM with the original introduction of the Token-Ring, is the Asynchronous Communication Server. The server (an IBM PC) provides access to/from the network via two (per server) switched lines connected to a modem or IBM/Siemens CBX.

An example network configuration is shown in Figure 5-3. The server requires NETBIOS for its operation and works with all IBM LANs

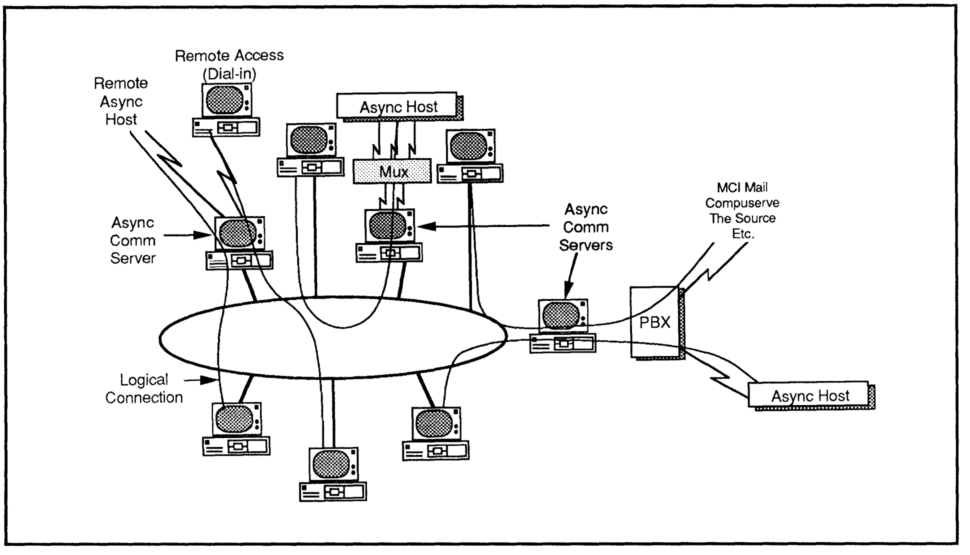

Figure 5-3: Asynchronous Communication Server

that support NETBIOS.

The server program runs in either a dedicated (executing the server only) or non-dedicated PC (executing the server + other applications). It allows specially programmed applications operating on the network to "dial-out" from the network, or specially programmed applications to "dial-in" to the network. The network must include a complementary application to service the in-coming call. It is important to realize the server is a program and a protocol specification; by itself, it won't do anything. For example, IBM does not provide a network terminal emulation package which runs on a workstation and communicates with the server. The user or third party must supply the necessary applications software. It is possible to set up two applications as "servers" and have them communicate with each other via the LAN. This is useful for debugging new communications server applications.

Third party vendors that have adapted their communications software to work with the server include: Hayes with SmartCom, Crosstalk Communications/DCA with Crosstalk and Software Publishing with PFS:Access.

Asynchronous Connection Server

The IBM Local Area Network Asynchronous Connection Server Program functions as an RS-232 communications server for Token-Ring or PC Network (broadband) providing access to asynchronous devices such as modems. The program provides functions similar to devices such as the Ungermann-Bass NIU (network interface unit) that supports serial ports or the 3Com Communications CS (communication server), both of which are available for token-ring operation.

The IBM version is different from U-B or 3Com in that it is based on a PC AT or the Personal System/2 Model 30. PCs communicate via the Token-Ring to the Async Server via an enhanced BIOS interface or via the Asynchronous Communications Server protocol. In addition to using the COM1 and COM2 serial ports, the server can contain up to four IBM Realtime Interface Co-processor Multiport cards with up to 8 ports per card for a total of 32 ports.

An interesting feature of the Async Connection Server is that devices attached to it, such as an IBM 3101 terminal, can seek a NETBIOS session over the Token-Ring to a second server that has asynchronous devices connected to it. Using this scheme, one can envision building terminal-only token-rings.

The Async Connection Server protocols are designed to handle data rates up to 64Kbps. The server uses the "AT" command set (used in Hayes Smartmodems and compatible modems). Handling high data rates may require a dedicated PC. One such flow control technique handled by the server is XON/XOFF.

The server relies entirely on NETBIOS and does not use any DOS services. The server runs in the background, relying on the PC's timer ticks (18.2/second) for its operation. The server can only service two lines (COM1 and COM2) but multiple servers can be attached. They then appear to an application as one large server. A server can automatically queue connection requests if all lines of a specified type are busy. Servers can be partitioned by function (i.e., a pool of 1200-baud modems) or group (only authorized groups can access a certain service).

The server protocol consists of a command set of fourteen commands to perform connection establishment, data transfer, and connection termination.

Connection establishment consists of the following commands: connection request (in which you can give the server a telephone number or a name to be looked up in a directory), connection request error (sent back if a connection request was to a unique (not group) name), connection identification and identification response (identify which server responded to the connection request), connection progress (your queue position), request queued (notification of non-immediate line access), and cancel queue (performed by the server -- this will happen if the line is lost, a number dialed is busy, not answered, etc).

Data transfer consists of the following commands: data (how much data the server is to send to a work station at one time -- e.g., character by character, a text line at a time, x number of characters, time-out between characters received, etc.), transmission status, query parameters, change parameters (specified by the data command), and connection parameters.

Connection termination is one command. With this command, an application can terminate its connection with the communicating device (such as a modem).

NETBIOS commands used for establishing sessions, hanging up sessions, and handling special network situations with the server include: call, listen, add name, add group name, cancel, and delete name.

NETBIOS commands used in conjunction with the server protocol commands are: send or chain send, send datagram, receive, and receive datagram. All are used to send protocol commands to the server.

Examples of command flows between and application and the server are shown in Figures 5-4 through 5-6.

Print Manager for 3820

The PrintManager is a program written to use NETBIOS which operates in an IBM PC attached to the Token-Ring or PC Network. When operating on the Token-Ring, the NETBIOS emulator and PC Network Program is required. The PrintManager provides the function of a print server specifically for the 3820 document printer. The printer is attached to an RS-232 interface on the PC, and operates at 19.2 Kbaud.

The PrintManager program includes over 50 (IBM-supplied) 240-by-240 dots-per-square-inch fonts, including representations of the IBM 5152 PC graphics printer. Other IBM PCs on the network can access the PrintManager from NETBIOS applications operating with the PC Network Program.

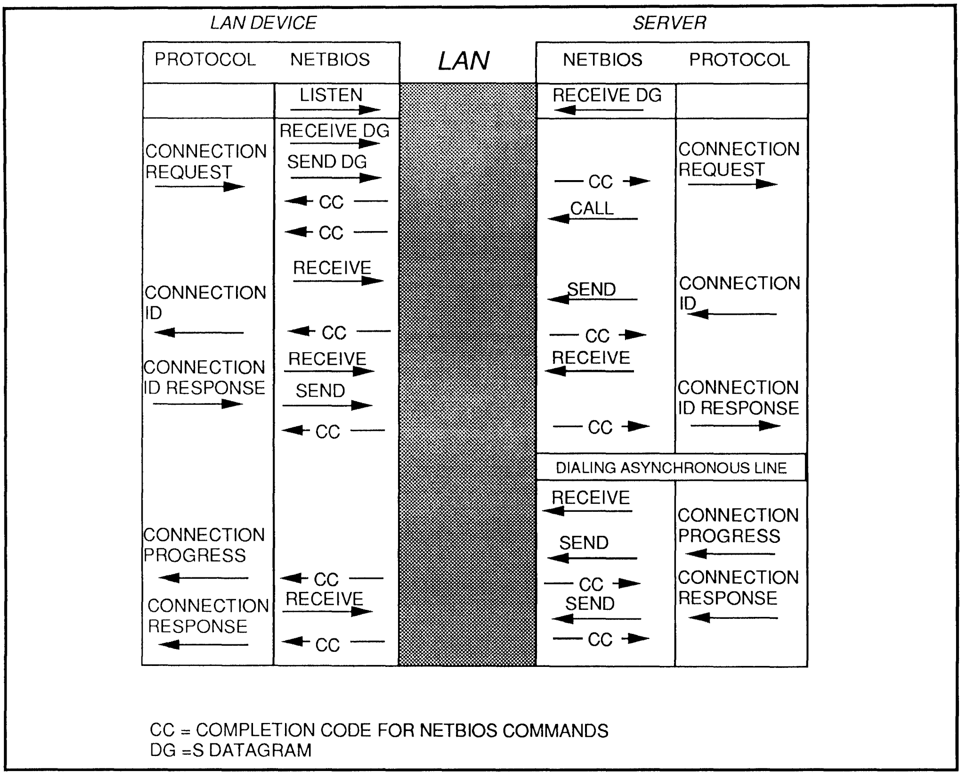

Figure 5-4: Outbound ACS Command Flow

The 3820 itself is a laser printer which provides cut-sheet duplex printing up to twenty pages per minute. Originally designed to operate as a remote printer in an SNA/SDLC environment, the Print-Manager allows it to operate with a PC attached to the Token-Ring.

Token-Ring/ PC Network Interconnect Program

The Interconnect Program allows a dedicated PC to act as a gateway between the Token-Ring and PC Network. A dedicated IBM PC running only the interconnect program is physically attached to the two networks with one token-ring adapter card and one PC Network adapter card. Applications written to NETBIOS can communicate with devices on either network.

As an example, an IBM PC LAN Program user can access programs or data on a server from one network to another. This requires the Interconnect Program to be "pre-configured" to identify the devices

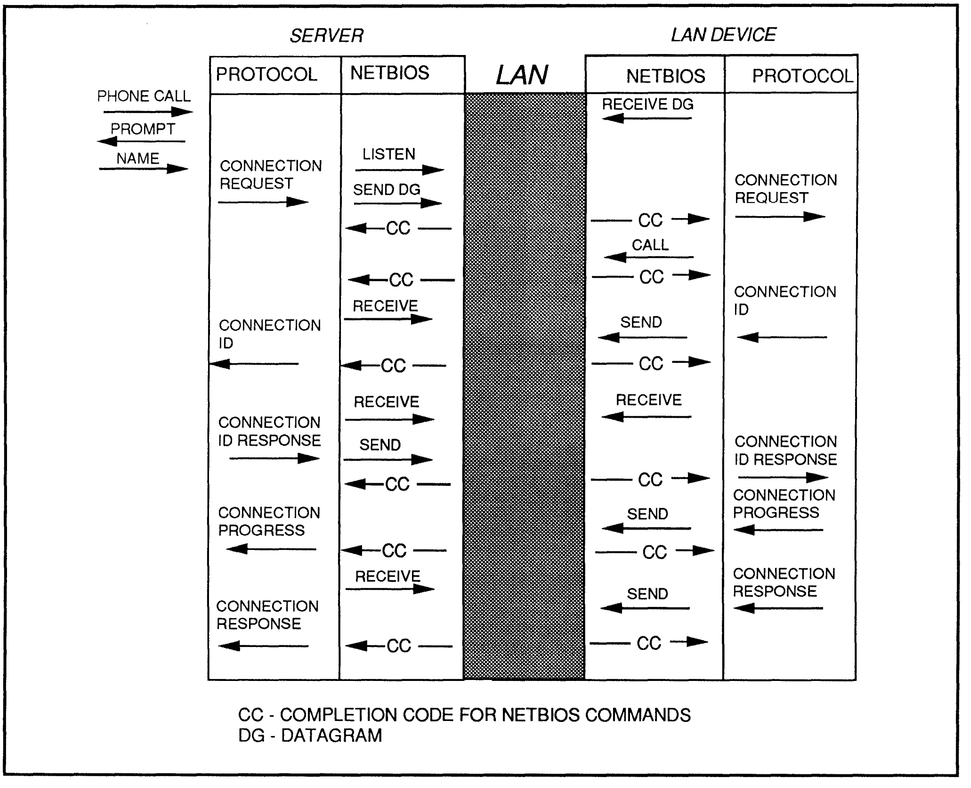

Figure 5-5: Inbound ACS Command Flow

(names) on each network which will be known to the other network. The names can not be dynamically changed during operation and only 16 names on each side are supported. During operation, the gateway receives messages from one network and forwards the messages to the other. Also, during operation, an operator can check device status and monitor the gateway activities. An example interconnection is illustrated in Figure 5-7.

3270 Emulation Program

The 3270 Emulation Program provides users on the Token-Ring access to IBM hosts from their workstations, without having dedicated coaxial wiring and SDLC cards on each workstation desiring host communication.

The 3270 Emulation program can be operated in one of three modes: 1) as a gateway on the network; 2) as a workstation on the network; or 3) as a stand-alone remote user station. The program was original-

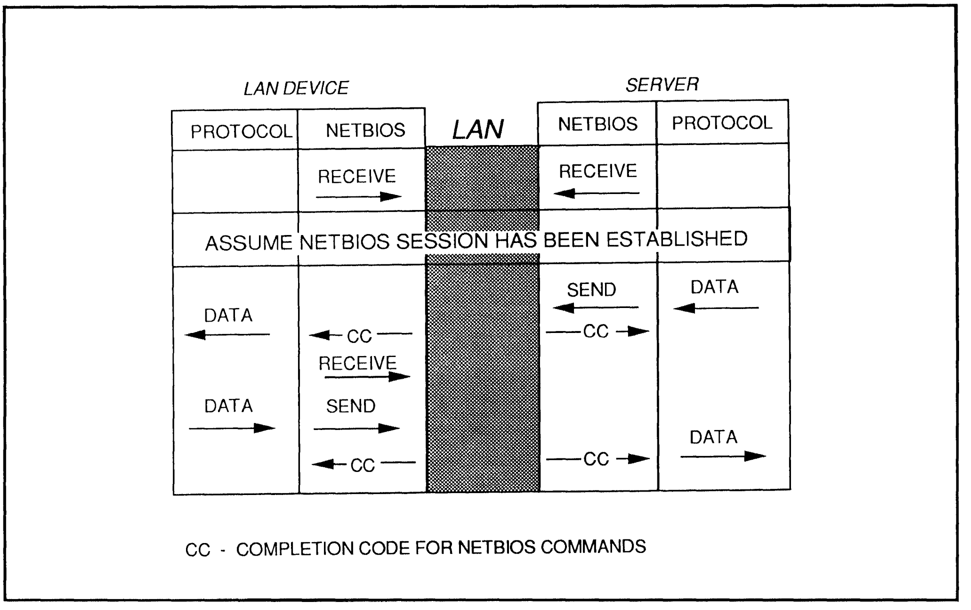

Figure 5-6: Data Transfer ACS Command Flows

ly written for the PC Network; thus it relies on NETBIOS for operation.

As a gateway on the network, the PC requires the IBM SDLC card (for "remote 3274 operation") or IBM 3278/3279 coaxial adapter board (for "Distributed Function Terminal (DFT) operation") to communicate with the host. The gateway serves users on the network who are running the program as a workstation. Up to 32 concurrent sessions are supported with SDLC; up to five sessions are supported with the coaxial attachment. Multiple gateways can be attached to the network, providing session access to a single host or multiple hosts. The gateway does not have to be dedicated, but dedication is recommended. IBM recommends the use of the PC AT as the gateway.

As a network workstation, the program uses the resources of the PC to emulate a subset of the IBM 3278-2 or 3279-S2A display station with an optional IBM Graphics Printer, Color Printer, Wheelprinter, or Quietwriter attached. The user can then establish a session with a host via the gateway. File transfer (with appropriate host software) to/from local disks or a file server's disk, screen save, and file append

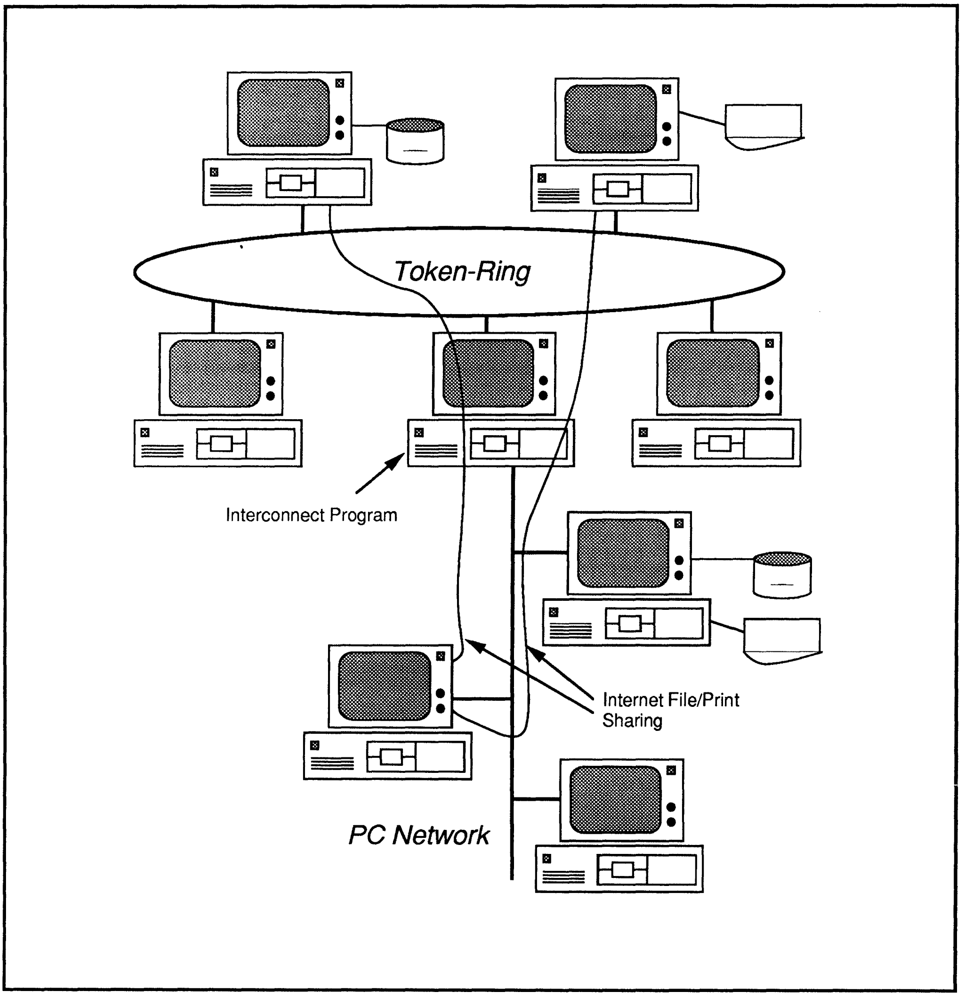

Figure 5-7: Interconnect Program

are supported. The user can also "hot-key" back and forth between the emulator and network operation.

Another workstation feature is the ability to have host or operator initiated printing to the workstation-attached printer (which mimics the operation of a 3287). The keyboard can be mapped allowing the user to redefine most keys on the PC keyboard to closely mimic the operation of the 3278 keyboard (such as PF keys mapped to the PCs function keys).

The workstation can communicate with a host via a PC configured as a gateway as described above, or via a Token-Ring-attached 3725 front-end processor (FEP) or 3174 cluster controller. An example setup on the Token-Ring is shown in Figure 5-8.

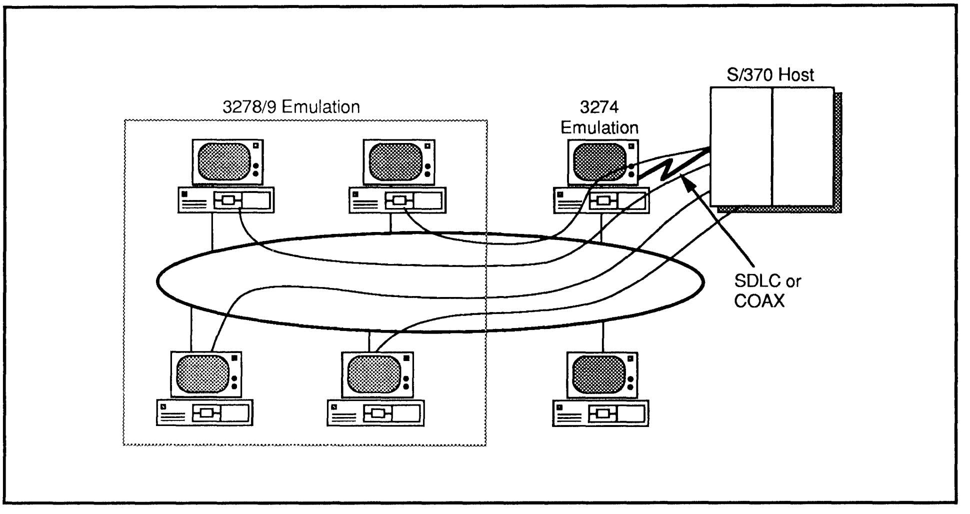

Figure 5-8: 3270 Emulation Program

The 3270 Emulation Program supports most features of a 3274 controller and 3278/3279 display station, but not all. Many functions such as structured field and attribute processing (SFAP); Programmed Symbols (PS) on attached terminals; extended color on terminals; extended highlighting on attached terminals; magnetic strip reader; selector pen; and security keylock are not supported.

3270 Workstation Program

The IBM 3270 Workstation Program version 1.1 allows direct communication to a Token-Ring-attached 3174 or 3725. The PC 3270 Emulation Program Entry Level Version 1.2 (for PCs and PS/2 Model 30) and Version 1.2 (for PS/2 Model 50 and higher -- yes, there are two versions of Version 1.2(!)) adds Enhanced Connectivity Facilities (ECF) to Version 1.1 by supporting the Server-Requester Programming Interface (SRPI). Although not a "turn-key" Token-Ring 3270 program, it offers the SRPI interface that can be used across the Token-Ring.

For more details, refer to the table shown in Figure 5-9 which summarizes the features of the IBM 3270 offerings for the Personal Computers and PS/2 family.

An IBM PC or an IBM Personal System/2 Running: May Access an Appropriately Configured Host by Connecting Via:	*IBM PC 3270 Entry Level Version 1.2*	*IBM PC 3270 Emulation Version 3.0*	*IBM 3270 Workstation Version 1.1*
IBM 3174/3274 (in CUT mode)	.Yes	No	Yes
IBM 3174/3274 (in DFT mode)	No	Yes	Yes
IBM 3276	Yes	No	Yes
IBM 4321, 32, 61	Yes	No	Yes
IBM 4701 Financial Controller	Yes	No	Yes
IBM Token-Ring Network and IBM 3725/3174	Yes	Yes	Yes
IBM LAN Program and PC Gateway	No	Yes	No

Figure 5-9: 3270 Options for PC

IBM Personal Communications/ 3270

IBM Personal Communications/3270, a DOS program, includes multiple 3270 display or print sessions, Emulator High level Language Application Program Interface (EHLLAPI), and the capability for concurrent connection to one or more IBM System/370* host systems as a 3270 terminal or workstation printer.

This program provides improved file-transfer performance and expands 3270 functions for workstations attached via a Token-Ring Network.

Expanded gateway services provide System/370 host connections to users of OS/2 Extended Edition, PC 3270 Emulation Program Version 3, 3270 Workstation Program Version 1.1, and Personal Communications/3270 via an IBM Token-Ring Network.

Series/1 PC Connect Program

Another application written for NETBIOS, the Series/1 PC Connect Program, provides a path between the Token-Ring (as well as PC Networks) and the Series/1, enabling workstations on the network to utilize Series/1 resources and communicate with large IBM hosts. The program requires an IBM PC attached to the network, the NETBIOS emulator, the IBM PC Network Program (configured as a server) and the Series/1 to PC Channel Attachment Feature. The channel attachment consists of a single "cycle stealing" card, a twelve foot intersystem channel cable, and a PC channel interface card. Programming support for the Series/1 is in the RPS operating system.

Figure 5-10 illustrates the use of the Series/1 PC Connect Program on the Token-Ring.

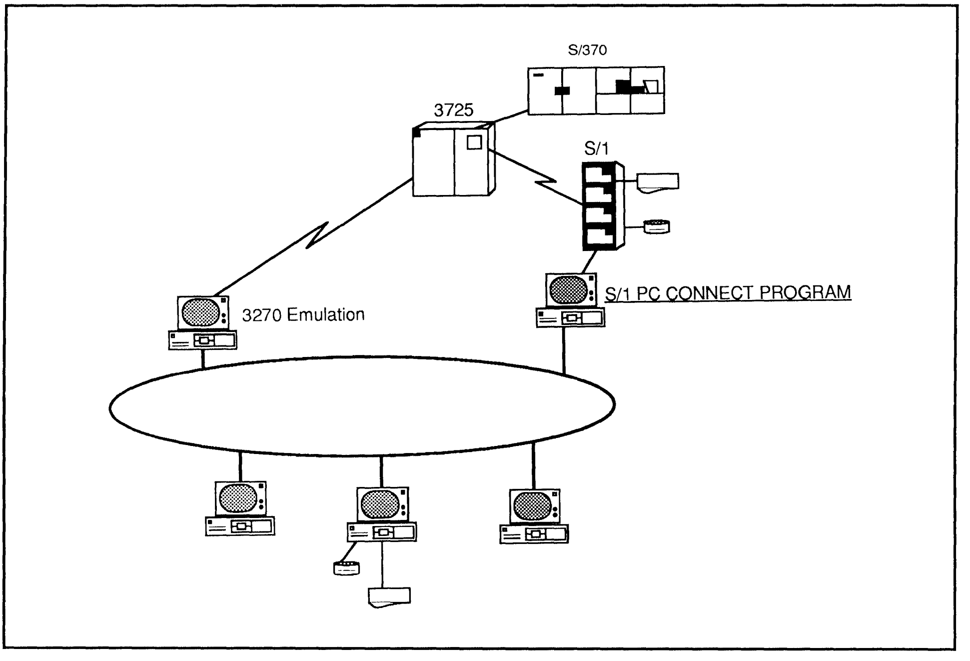

Figure 5-10: Series/1 Connect

The PCs in the network can use the Series/1 disks and printers as though they were disks attached to the server. They can also use the Series/1 as a "pass-through" communications controller to communicate with large hosts, using a variety of Series/1 BISYNC or SDLC communications options. The host communication option requires the PC 3270 Emulation Program to operate in a separate PC (the file server and the emulation program cannot operate in the same PC).

System/36 Support A System/36 Model 5360 or 5362 can be attached to the Token-Ring with the IBM System/36 Local Area Network (LAN) Attachment feature with the IBM System/36 PC 5360/5362 LAN Communications Licensed Program and the Model 5364 with the IBM System/36 PC 5364 LAN Licensed Program. All models can connect to a maximum of two rings.

The 5360/5362 attachment feature consists of a direct-attach System/36-to-AT adapter that is installed in a dedicated AT that in turn is connected to the Token-Ring via Adapter II. For the 5364 (also known as the System/36 PC), the AT is directly attached. For either configuration, the communications program is downloaded into the AT from the System/36.

With the introduction of the low-end System/36 Model 5363, IBM is offering a direct channel attached Token-Ring adapter. This adapter will eventually be available for larger processors in the System/36 family.

The PC can make use of System/36 resources through PC Support/36 or Personal Services/PC. PC Support/36 allows the user to transfer PC-DOS or System/36 files back and forth, with or without translation, to access the System/36 in terminal emulation mode, or to access disks and printers on the System/36 as virtual disks and printers known to PC-DOS.

Personal Services/PC is more of an office automation package that allows users to exchange files and electronic mail via System/36 and DISOSS (which operates on System/370 hosts). Versions of Personal Services are also available for System/36 and System/370 users.

TCP/IP

In April of 1987, IBM quietly announced a number of new TCP/IP implementations (its first announced TCP/IP product was for the RT PC with an Ethernet adapter). The IBM 8232 LAN Channel Station allows PCs on the Token-Ring, PC Network Baseband, PC Network Broadband, Proteon's proNET line, and most Ethernet implementations to communicate with a mainframe using TCP/IP.

IBM TCP/IP for the PC allows PCs to connect to System/370 hosts running IBM's Virtual Machine (VM) operating system. PCs directly attached to the Token-Ring can communicate via PC-DOS TCP/IP to a directly attached 9370 (currently the only System/370 computer with a direct plug-in Token-Ring adapter). The same holds for Ethernet. With this TCP/IP capability, IBM is providing what it terms "OEM connectivity" (read multi-vendor connectivity) for the 9370. In fact, an environment such as the one depicted in Figure 5-11 can be configured so that one could, for example, operate a 3720 terminal emulation program on a Sun Workstation attached to an Ethernet to communicating via an RT PC gateway a 9370 on a Token-Ring. Then the 9370 operating VTAM under VM, converts the TCP/IP protocol and sends SNA data back through the Token-Ring, to another 9370, on to a 3090. Data sent from the 3090 back to the workstation follows the same path in reverse. A word of caution -- the level of 3270 support from other vendors via Ethernet varies dramatically (for example, DEC only supports line mode, not full screen mode).

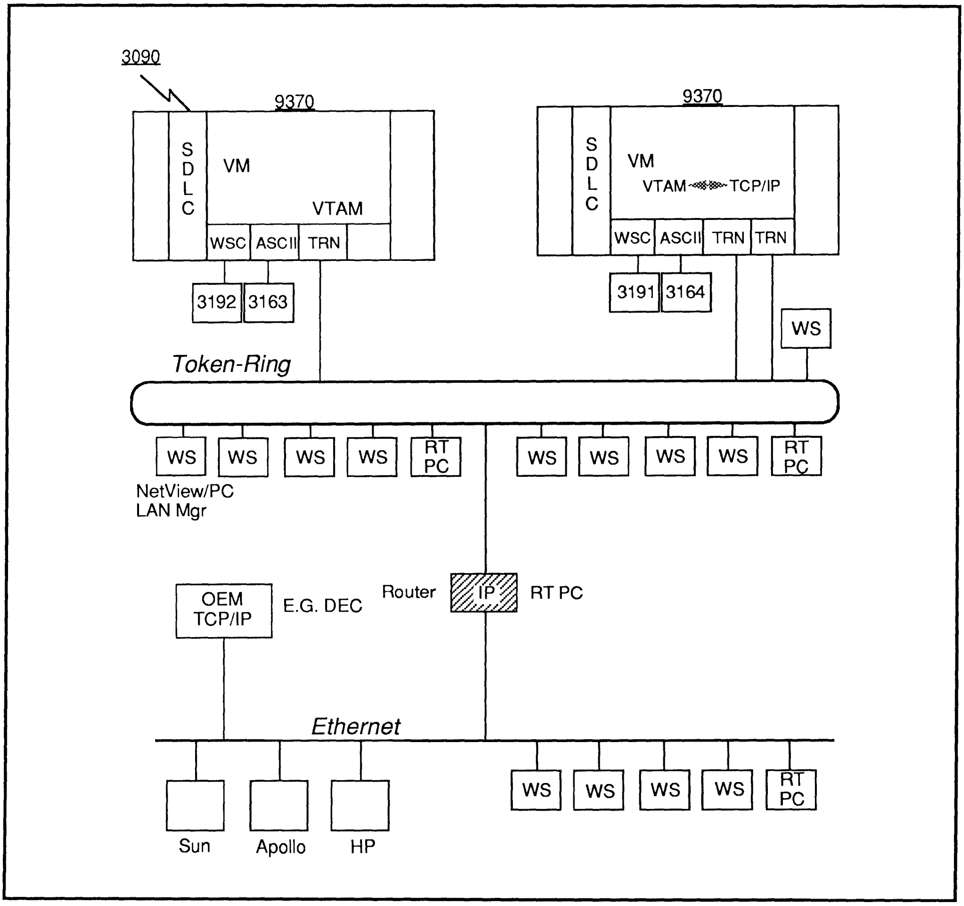

Figure 5-11: 9370: Ethernet to Token-Ring via TCP/IP

Two more IBM TCP/IP products include the Series/1 X.25 DDN and the 7170 Data Acquisition Control Unit: the Series/1 X.25 DDN can be used to interconnect TCP/IP LANs, and the 7170 can be used to communicate to IBM 43xx, 308x, and 309x mainframes via Ethernet or proNET.

Chapter 6

Operating System/2

Operating System/2 is the "new" operating system developed with Microsoft that was announced in April of 1987 with IBM's family of Personal System/2 computers. This was IBM's opportunity to offer its first Systems Application Architecture (SAA) operating system.

IBM offers their own extended version of OS/2 called Operating System/2 Extended Edition. One of IBM's unpublicized goals of OS/2 EE is its intent to forge strong links with System/370 environments.

OS/2 is the primary operating system for IBM Personal System/2 with the Micro Channel architecture and 80286 or 80386 processors. This includes the Models 50 through 80, and existing PC ATs and the XT/286. OS/2 will support any future IBM Personal Systems based on 80286, 80386 or 80486. OS/2 runs in the protected mode of the 80286 and functions likewise in the 80386 or 80486. Thus, it does not take advantage of additional features available in the 80386/80486 such as virtual DOS machines or large memory segments for programming.

It does however, give users and programmers capabilities not previously available under PC DOS and 8088/8086 processors (such as running programs larger than 640 Kbytes and supporting concurrent multiple applications). OS/2 does support PC-DOS in a "compatibility box" for many (but not all!) existing programs.

OS/2 also takes advantage of the VGA display, which was new with the PS/2 (VGA is now the defacto IBM graphics standard for its PCs). The VGA standard and PS/2 standard keyboard are important hardware contributors to the consistency requirements of SAA.

OS/2 standard edition (or the base operating system) does not contain the communication and database managers: users may install the standard edition and later migrate to the extended edition if communications and database support is desired.

The capabilities of OS/2 will be briefly examined below, followed by a closer look at the communications capabilities of the extended edition and OS/2 server software available from IBM and Microsoft.

OS/2 Features

OS/2 is a multi-tasking, multi-threaded operating system which breaks the PC DOS 640-Kbyte RAM barrier, increasing memory addressability to 16 Mbytes; it also provides a base for larger, functionally richer applications, and processing capability for greater amounts of data. With OS/2, multiple programs may reside in memory and concurrently share system resources, such that each application may also perform multi-tasking within itself. Programs may also take advantage of the 80286/386 protected mode, which restricts cross program interference. Another feature of OS/2 is support for virtual memory -- programs may be segmented to allow them to run in memory-restricted environments with swapping to/from a hard disk.

OS/2 also improves on the user and program interfaces. There is a new program selector, installation aid, a familiarization program, error handling, and on-line help facilities. Besides text, there is also a graphics and window interface for not only the operating system but application programs as well. Included are high-level application program interfaces (APIs) to provide developers with common interfaces for intelligent workstation programs. This will lessen the impact from future operating system releases, new system units, and new peripheral devices. The API idea follows the design philosophy of computers like the Apple Macintosh, in which many of the functions (such as user interface, menus, etc.) are imbedded into the OS, so that software developers do not have to concern themselves with developing that part of the application. Of course, this promotes user interface consistency from application to application. This is one of the key strengths of the operating system of Macintosh that is now entering into the IBM PC world. In addition, OS/2 retains the familiar command line commands of PC DOS for experienced users, and provides new commands to take advantage of extended functions.

OS/2 Extended Edition provides an extended system software solution consisting of the base operating system, (i.e., the standard edition) and the OS/2 communications manager and database manager. The OS/2 communications manager provides host connect and local area network support; supports concurrent hosts with both network and local workstation activities; supports emulation of multiple terminal types; supports transfer of files between PC and host or a network server; provides application program interfaces; and provides communications and system management support.

The OS/2 database manager provides both a direct user interface and application program support via commands, prompts, and procedures. It also provides utilities to aid in the creation of data entry and query screening, menus, and reports. Additionally, sorts and calculations can be performed. OS/2 participates in SAA's definition for database languages Structured Query Language (SQL), also used on DB/2, SQL/DS, and QMF host products. In addition, a language interface to the IBM C/2 language compiler is provided. The database manager provides utilities to import data files created in DBASE III, Lotus 1-2-3, Symphony, or the IBM Personal Decision Series (PDS) formats. The database manager can also present data in the form of tables, using columns and rows to select and display information, and data can also be retrieved from multiple tables. OS/2 database manager supports queries of relational data stored at a workstation, locally, or remotely on a server.

The various releases, or versions, of OS/2 are summarized here: OS/2 standard edition 1.0 provides the base operating system, without the communications manager and the database manager; Version 1.0 is essentially an early release of the standard edition that contains only a text user interface; OS/2 standard edition 1.1 adds the graphics/windowing interface (also known as the presentation manager); OS/2 extended edition 1.1 adds the communications manager, and the database manager -- the base operating system is the same as Standard Edition 1.1. Either version 1.1 also supports fixed disk partitions greater than 32 Mbytes.

The presentation manager provides more than graphics support for the CUA aspects of SAA (see Chapter 2). For example, a cut-and-paste graphical data exchange called the picture interchange is provided. The presentation manager supports both vector and bitmap graphics.

Communications Manager

The Communications Manager provides communication services for applications written for the IBM Operating System/2 environment, between IBM PC and IBM Personal System/2 and host networks. These services provide communication to personal computers and systems over a wide range of local and remote connectivities -- including SDLC, Distributed Function Terminals (DFT) mode to an IBM 3174 or 3274, IBM Token-Ring Network and IBM PC Network, and asynchronous links -- by utilizing LU6.2, 3270 data stream (LU2), and asynchronous communications protocols. Emulation support for multiple terminal types concurrently is provided, and file transfer and a keyboard remap facility is supported. Several programming interfaces are provided to allow programs to take advantage of the power of the IBM PC and IBM Personal System/2 and to facilitate programmer productivity in applications development. The Communications Manager provides alerts for network management functions for problem determination, and controls for SNA communication services.

Data Link and Data Stream Support - An IBM PC or IBM Personal System/2 may be attached to an IBM PC, an IBM Personal System/2, a host, or departmental system, locally via the IBM Token-Ring Network or IBM PC Network and DFT (to an IBM 3X74 controller), and remotely via SDLC and asynchronous links.

Programming Interfaces (PIs) - The Programming Interfaces (PIs) supported by the Communications Manager include:

Advanced Program-to-Program Communications (APPC) PI: The LU6.2 architecture describes the functions that may be used by conforming pairs of programs for Advanced Program-to-Program Communications over the supported data links. The interface provides programming access to these functions (or verbs). Support is for both Mapped (data stream independent) and Basic (data stream dependent) verbs. APPC applications may be written to IBM hosts (with MVS-CICS and VSE-CICS), IBM System/36, IBM System/38, AS/400, IBM Personal System/2, IBM Personal Computer, IBM RT PC, IBM System/88, and IBM Series/1 systems.

Server-Requester Programming Interface (SRPI): This is the PI for the Enhanced Connectivity Facilities. It enables the writing of simple, communications-independent, requester programs which can call to host server programs, with synchronous returns. It is supported over links using LU2 protocols. Host server support is available under MVS/TSO and VM/CMS.

Asynchronous Communications Device Interface (ACDI): This interface allows the writing of applications (such as other asynchronous emulators or file transfer programs) to exchange data over asynchronous links. The interface provides a high degree from independence of the asynchronous hardware used. Device-specific programming modules are required for each supported device type and are included in the product. They are transparent to user applications. Supported functions include the ability to manipulate the line characteristics and connection control (connect and disconnect) without having to deal with physical device-specific characteristics.

IBM Local Area Network PIs: The IBM NETBIOS and IEEE 802.2 Data Link Control PIs are provided for communicating across IBM LANs.

Programs written for the Communications Manager may invoke the supported communications functions by calls from IBM Pascal/2, IBM Macro Assembler/2, and IBM C/2.

Terminal Emulation - The Communications Manager allows concurrent emulation of synchronous and asynchronous terminals. Emulation includes IBM 3270, IBM 3101, and DEC VT100 terminals.

IBM 3270 Terminal Emulation: The following terminals can be emulated: IBM 3178 Model 2; IBM 3278 Models 2, 3, 4, and 5; IBM 3279 Models S2A and S2B. The following are supported: all base data stream functions; multiple interactive screens; extended attributes; extended data stream (including seven colors, and extended highlights); file transfer; emulator keyboard remapping; LU Type 2, node Type 2.0, to a maximum of five 3270 display sessions per workstation over SDLC, IBM Token-Ring Network, and DFT links.

IBM 3101 and DEC VT100 Terminal Emulation: An IBM PC or an IBM Personal System/2 connected to a host supporting an asynchronous link can emulate the IBM 3101 (Model 20) or the DEC VT100 terminals. Lines can be switched, non-switched, or direct-connected and must be compatible with 1984 CCITT V24/V28 (RS-232-C) as implemented by IBM.

File Transfer - The Communications Manager supports the following file transfer types between supported hosts and IBM Personal Computers and IBM Personal System/2: IBM host file transfer programs are supported under 3270 and asynchronous emulation, (Under asynchronous emulation, file transfer with the 3270-PC File Transfer Program includes four-byte CRC error detection); support for XModem is under asynchronous emulation with 128-byte block transfer and one-byte checksum error detection; and Pacing under asynchronous emulation with line delay interval pacing or host prompt character pacing is supported in sending an ASCII text file.

Communications and Systems Management - Communications and Systems Management (C&SM) support (for IBM System/370 host network) includes: C&SM alerts for SDLC, ASYNC (requires a SDLC or IBM Token-Ring Network link to communicate alerts to the host), IBM Token-Ring, and IBM PC Network data links, and problem determination data.

Problem Determination - The Communications Manager provides functions for gathering and processing problem determination data. These functions include tracing of programming interfaces, data units, and/or system events; displaying and printing of all or selected error logs from file; system dumping; and displaying of all or selected messages.

Subsystem Management - The Communications Manager allows a user's system administrator to control and obtain status information on the SNA communication resources maintained by the Communications Manager. As a management tool, it displays information on which programs are in use, sessions in use by the programs, detailed information about the sessions, and resources which are active. It allows the activation or deactivation of sessions, data link controls, and specific links. It also can be used to start and stop an attach manager which allows remote applications to be started.

Figure 6-1 summarizes the various interfaces available under the communications manager for the token-ring.

Enhancements

OS/2 Standard and Extended Editions Version 1.2 have significant new functions, including many features that better conform to SAA. 1.2 includes enhanced communications functions that allow better connectivity to IBM mainframe computers, multi-user support for remote data services, and support for additional programming languages.

An integral part of OS/2 Standard Edition Version 1.2 is a Desktop Manager/File Manager -- an enhanced user interface to operating system functions such as adding and starting applications, associating files to applications, and switching between applications. The interface also can make use of icons.

OS/2 Standard Edition Version 1.2 also contains the new High Performance File System (HPFS) that may be used to replace the PC-DOS and earlier OS/2 File Allocation Table (FAT) file system. The

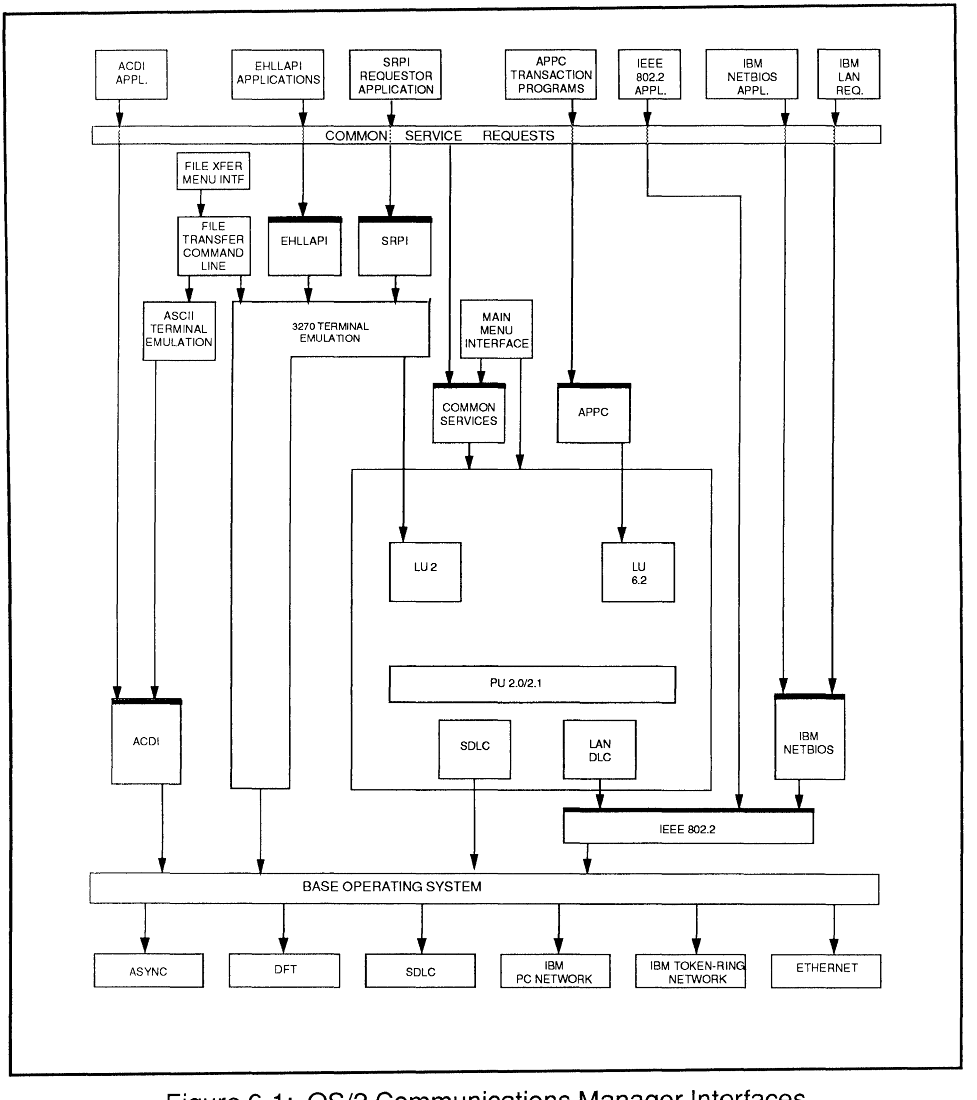

Figure 6-1: OS/2 Communications Manager Interfaces

HPFS can handle partitions and files up to two Gbytes in size and delivers improved performance over the FAT file system. Compatibility between files created by both file systems is maintained for OS/2 programs, as well as for DOS programs running in OS/2's DOS compatibility mode.

OS/2 Extended Edition 1.2 (which serves as the development platform for IBM's new OfficeVision/2 LAN series), has enhanced its Database Manager with Remote Data Services (RDS), an SAA function that provides multi-user database support in an OS/2 LAN environment. RDS gives workstations, attached on a LAN, access to databases on other workstations on the same LAN, without requiring the user to know, or identify, on which workstation the information resides.

Other Database Manager enhancements include: referential data integrity, which helps ensure consistency of data within a database; cursor stability; Cobol, Pascal, and Fortran precompilers; and grant/revoke authorization capability.

A new feature of the OS/2 Extended Edition Communications Manager is an SNA Gateway. This LAN-based Gateway allows customers to support up to 254 workstations on a single LAN. Previously, a maximum of 32 workstations could be supported on a single LAN with IBM's PC 3270 Emulation Program.

Also, the Communications Manager utilizes the Network Driver Interface Specification (NDIS) for two additional LAN protocols: Ethernet DIX Version 2.0 and IEEE 802.3. This support provides the ability for NETBIOS, LU2, and LU6.2 functions to be used across an Ethernet LAN connection.

Other new communications features include a 5250 Work Station Feature; host-directed print; support for the X.25 protocol; and host-graphics view. In order to use the host graphics, GDDM-OS/2 Link is required.

The Database Manager's Query Manager and the Communications Manager's 3270 and ASCII terminal emulators are now Presentation Manager-based.

OS/2 Extended Edition's new SAA Procedures Language is a general-purpose, interpretive language suitable for use as a command processor, a macro language, and a programming language for the casual programmer. It combines the structured logic, general variables, and subroutine calls of a traditional programming language with the ability to execute character strings as system commands.

IBM's OS/2 LAN Application Programming Interface (API) support has been extended to include new APIs, such as Files, Messages, and Sessions. This allows customers and software developers to create distributed requester/server applications that will operate in an OS/2 LAN server environment.

Microsoft LAN Manager

At the same time that IBM announced OS/2 and OS/2 LAN Server, Microsoft announced the Microsoft OS/2 LAN Manager server software (not to be confused with the IBM LAN Manager, a network management tool). The Microsoft OS/2 product is the first to be announced as the result of the joint development agreement announced by IBM and Microsoft in August 1985. Microsoft is offering MS OS/2, including the MS OS/2 Windows presentation manager, to all its existing OEM customers.

The MS OS/2 LAN Manager (developed in conjunction with 3Com) allows users to link personal computer systems running either MS OS/2 or MS-DOS together on a single network using token-ring or Ethernet. Systems running Microsoft XENIX and XENIX Net may also be connected to the same network.

An MS OS/2-based system may act simultaneously as a workstation and a server within such a network; MS-DOS systems running Microsoft Networks can act either as a server or a workstation. XENIX-based systems may act as both server and workstation simultaneously.

The MS OS/2 LAN Manager provides networking capabilities including transparent file and print sharing, user security features, and network administration tools. Because the MS OS/2 Lan Manager is tightly integrated with the MS OS/2 operating system, MS OS/2 programming interfaces work transparently across the network. The interprocess communication (IPC) capability facilitates applications that can communicate directly with each other, even though they are resident on different machines on the network. Application developers will be able to write a single version of a software product which will be capable of running either on a single machine, or distributed between machines connected by the MS OS/2 LAN Manager. Vendors such as Novell and 3Com have offered their own versions of IPC prior to OS/2.

The MS OS/2 LAN Manager is fully compatible with the older Microsoft Networks products for both the MS-DOS and XENIX operating systems. This allow systems running MS OS/2 to be integrated into existing networks supporting MS Networks with minimal disruption to the users of the network.

Most OEMs that offer MS Networks will also be offering the MS LAN Manager. IBM is the notable exception. Even though MS Networks and the PC LAN Program/NETBIOS have a lot in common (MS Networks now incorporates a NETBIOS emulator), IBM did not include in its original announcement support for the LAN Manager opting later to offer the OS/2 LAN Server software (announced in November of 1987). The other major difference between Microsoft and IBM is the IBM OS/2 Extended Edition -- this appears to be a "Value-added" version of OS/2 by IBM that Microsoft will not offer. It will be up to the OEM's of MS OS/2 to decide whether or not to add support for IBM's Extended Edition. Many vendors will offer add-ons to Standard Edition, such as an APPC/PC package.

In November of 1989, Microsoft announced LAN Manager 2.0. Among a list of major enhancements, LAN Manager 2.0 includes:

- 386/486 microprocessor support
- OS/2 High-Performance File System (HPFS)
- Multiprocessor support
- Facilities that let multiple servers be administered as a single server
- Tighter security at both the workstation and server
- Fault tolerance
- Reduced memory requirements for MS-DOS and PC-DOS operating systems workstations
- Peer services
- An improved user interface
- Easier installation

The 80286, 386, and 486 all implement a hierarchical memory-protection scheme, simplest to picture as a series of concentric rings. The innermost ring, Ring 0, is the most highly privileged, and the outermost ring, Ring 3, the least privileged. Typically, highly reliable and trusted code, such as the operating system kernel and device drivers, run at Ring 0. Programs running at Ring 0 have direct access to all hardware devices and operating system functions.

Less-trusted code, such as application programs, run at Ring 3, the outermost ring. The hardware on the Intel chips ensures that application code running at Ring 3 cannot directly access or interfere with code or device drivers running at Ring 0. Thus the operating system is protected from any bugs that may be present in application-level code.

On 80286-based server hardware, the MS LAN Manager server runs as a Ring 3 OS/2 application program, uses the Microsoft OS/2 Version 1.2 High-Performance File System (HPFS), and performs all network I/O through standard MS OS/2 Version 1.2 kernel services. This provides adequate performance for small- to medium-sized networks where network I/O requirements are not too demanding.

For maximum performance on 386- or 486-based hardware, LAN Manager Version 2.0 automatically installs a network I/O subsystem designed to utilize the 32-bit instructions and 32-bit-wide data path of the 386 and 486 processors. This subsystem consists of 386-specific OS/2 kernel extensions and a 386 version of the MS OS/2 Version 1.2 High-Performance File System (HPFS-386).

The special network I/O subsystem for 386/486-based servers consists of an optimized Ring 0 server message block (SMB) server tightly coupled with a bootable, installable file system (HPFS-386), which run together as an MS OS/2 Version 1.2 File System Driver (FSD).

The Ring 0 SMB server performs its own internal thread management and scheduling to optimize the handling of network tasks. It runs alongside the standard MS LAN Manager Ring 3 server and works closely with the Microsoft NETBIOS transport stack and HPFS-386. It is not a substitute for the Ring 3 server; rather it provides special-case, enhanced performance for remote access to HPFS-386 volumes. So that the Microsoft LAN Manager Version 2.0 Ring 0 server can be optimized for the most common and demanding operation--namely file I/O--any requests destined for non-HPFS-386 resources, such as the DOS file allocation table (FAT) file system, character devices, and named pipes, are passed by the Ring 0 server onto the LAN Manager Ring 3 server, which satisfies the requests through OS/2 APIs (application programming interfaces).

LAN Manager Version 2.0 servers need not be dedicated to exclusive use by LAN Manager but can also serve as platforms for supporting server applications such as the Ashton-Tate/Microsoft SQL Server and the DCA/Microsoft Communications Server (containing many of the functions found in the IBM OS/2 Extended Edition.) Eventually, however, IBM will offer it Database Manager, Communications Manager, and LAN Requestor as separate OS/2 offerings.

The Microsoft LAN Manager Version 2.0 API set is a superset of the MS LAN Manager Version 1.0 APIs. All existing software written to these APIs will function identically when accessing a LAN Manager Version 2.0 286- or 386-based server, without change or update.

Network security in LAN Manager Version 2.0 has been strengthened at both workstation and server levels to help prevent unauthorized users from accessing and modifying data. Specific security measures include password aging, time- and workstation-specific restrictions, and a password validation delay (effective at blocking password-finding programs). Security on 386/486 servers is strengthened considerably: no on can log on to the server itself without administrative privilege, and no one can circumvent server security by rebooting the machine.

Microsoft claims that LAN Manager Version 2.0 is the "total systems approach to network server software." While many conventional networking solutions have focused on proprietary network or file system technology, MS LAN Manager Version 2.0 focuses on eliminating bottlenecks, from the network to the file system to the disk system.

With LAN Manager Version 2.0, Microsoft is finally enabling OEMs (such as 3Com) to enter the PC LAN file server mainstream to compete directly with heavyweights Banyan and Novell.

Microsoft is hoping the new LAN Manager will be a success and has lined up numerous OEMs, including AT&T, Apricot, Compaq, DCA, NCR, Nokia Data, Olivetti, 3Com, and Ungermann-Bass.

IBM LAN Server

The IBM OS/2 LAN Server provides, to both DOS (via the PC LAN Program) and OS/2 applications, resource sharing for disks, printers, and serially attached devices, plus facilities for defining, controlling, and managing access to LAN resources, as well as security and print management (for up to eight printers). These security and administration features are similar to those provided by PC LAN Program Version 1.3 except that file security is down to the file level. The LAN Server requires OS/2 Extended Edition and utilizes the OS/2 multi-session and caching functions. LAN Server also provides facilities for remote execution of programs.

Portions of the LAN Server have been licensed from Microsoft, most notably the redirector. IBM has not implemented many of the Microsoft Application Program Interfaces (API) in LAN Manager, most notably interprocess communication facilities that are inconsistent with Systems Application Architecture (such as Named Pipes --

a facility found in Microsoft's LAN Manager and later added to IBM's LAN Server). IBM and Microsoft are said to be working toward a common server in future releases.

IBM expects one to use the APPC/PC included with OS/2 Extended Edition to develop distributed applications -- especially in mixed computer Token-Rings. For strictly PC LANs, the issue is not as critical -- one is probably safe writing to NETBIOS or Named Pipes. Major third-party LAN vendors such as 3Com, Banyan, and Novell, are offering at least some level of compatibility (such as APPC) with OS/2 Extended Edition and OS/2 LAN Server and will increase this level of compatibility over time.

The newest OS/2 LAN Server (Version 1.2) has added support for Ethernet.

Highlights of Version 1.2 include:

- DOS LAN Requester and LAN Support Program -- Requesters allow individual workstations to access the server on a local-area network. The OS/2 LAN Requester continues to be included in the OS/2 Extended Edition operating system. For the first time, the OS/2 LAN Server includes the function previously provided by two products required for the DOS Requester. Moreover, in many PS/2s with extended or expanded memory, the DOS LAN Requester will allow 10-35 Kbytes more space for applications.
- OS/2 LAN Server Version 1.2 Performance Advantages -- By using the High Performance File System (HPFS) capabilities of OS/2 Extended Edition 1.2, file server performance and the number of files available for concurrent access by users on the LAN are increased.
- User Profile Management -- This capability permits workstation users to access both the LAN Server and the OS/2 Extended Edition Version 1.2 with a single user ID and password.
- File Replication Service -- This function allows critical files to be replicated from one OS/2 server to another OS/2 server or OS/2 requester on a time-interval basis for file protection and data integrity.
- New Application Programming Interfaces -- IBM's OS/2 LAN Server 1.2 application programming interface (API) support has been extended to include new APIs. These APIs allow programmers to create applications for network management, network messaging, shared resource management, LAN interprocess communications, and security.

The mission of LAN Server is a critical one for IBM -- IBM is hoping that it will be a key product to provide the means to migrate from DOS to OS/2. One of the stumbling blocks to achieving this goal is the stiff requirements and expense to operate IBM's OS/2 on the IBM token-ring.

Operating IBM's OS/2 on the token-ring requires the Communications Manager found only in OS/2 Extended Edition. So one pays the extra cost (for each workstation) of all of the features found in OS/2 EE just to operate it over the token-ring. As noted previously, IBM will apparently unbundle the Communications Manager and LAN Requester to allow other OS/2 vendors as well as users of the OS/2 Standard Edition, to operate on the token-ring (as well as Ethernet) without having to purchase OS/2 EE. This will help to accelerate the acceptance of OS/2 and OS/2 LAN Server/LAN Manager.

Chapter 7

Management

Management is the act of administering to the needs of the network. This involves overseeing the installation/ attachment and movement of devices, changes to cabling, performing diagnostics and recovery procedures in case of catastrophic failure, expanding the network, and determining the needs of users when installing the system to ensure proper accessibility and operation.

Once the system is operational, tools are needed to keep the system running smoothly and efficiently. When the Token-Ring was first announced, there were no such tools. Since that time, IBM has substantially beefed up its management offerings. In addition, third-party products are becoming available to complement the IBM offerings.

Installation

The two major phases in installing the Token-Ring are the cabling phase and the device phase.

There are four basic steps in installing a cabling system: planning, ordering, installing, and testing. Planning is the most crucial step since a bad plan will "ripple" through all other installation steps.

To plan cabling for a token-ring, one must understand all of the components of the system. The different cable types were discussed in Chapter 2, and are summarized in Figure 7-1. Cable choice will depend primarily on building codes, distances, and whether or not the cable will accommodate voice (telephone) service in addition to token-ring data. Voice service accommodation will also dictate which faceplates are used in the user work area.

RECOMMENDED USES OF CABLE	CABLE TYPES							
	1	1 OUTDOOR	1 PLENUM	2/3	2 PLENUM	5*	6	8
INTERCONNECTION OF WORK AREAS AND WIRING CLOSETS	X		X	X	X			
DATA COMMUNICATION BETWEEN WIRING CLOSETS	X		X			X		
DATA ONLY	X	X	X			X	X	X
DATA AND TELEPHONE				X	X			
OUTDOOR AERIAL AND DRY UNDERGROUND CONDUIT INSTALLATION		X				X		
PATCH AND EXTENSION CABLES							X	
INSTALLATION INSIDE CONDUIT OR ENCLOSED WIREWAYS	X			X		X		
INSTALLATION (WITHOUT CONDUIT) IN PLENUMS			**		**			
OPTICAL FIBER COMMUNICATION						X		
DATA COMMUNICATION UNDER CARPETED FLOOR								X

* Can be used both indoors and outdoors

** Check local electrical code to determine if installation is allowed

X = YES

Figure 7-1: Cabling System Summary

If one is installing outdoor Type 1 wire, IBM strongly recommends the use of surge suppressors to minimize damage from voltage surges. Both indoor and outdoor models are available. Caution must be used due to the possibility of an excessive ground potential difference between two power services' grounding systems. This could be a hazard between two buildings that are powered by different service entrances (multiple transformers), or could occur in a large facility. A test can be performed to determine ground potential difference; if it exceeds 1.0 volt AC, an engineer or the power company must be consulted to rectify the problem.

A good place to begin planning is the work area. A survey of the types and quantities of terminals or PCs should be completed to determine which accessories are needed. The work area location will determine the cable type, version (plenum, non-plenum, or outdoor), and length (from the wiring closet) needed. Cable choice will determine the type of support system needed.

Concurrently, with determination of the work area requirements, one should identify potential wiring center sites. Trade-offs can be made between one or two large centralized wiring centers or several smaller, decentralized wiring centers. While fewer wiring centers can lead to easier maintenance of the cable, it may also require larger amounts of cable. An example of two wiring closet placements is shown in Figure 7-2.

If the system is small, wiring closets may not be necessary. A system could be made up of one or two MAUs with patch cables.

Once all components are determined, they can be ordered (all components except the raw cable) through IBM or a third party independent contractor. Several third party contractors also offer site survey and planning services.

Part of the planning process is to identify (label) all components in the cabling system. The administrator will keep a set of charts and tables identifying the various components. An example labeling scheme, recommended by IBM, is given in Figure 7-3. Cabling components are identified by an eight-digit number consisting of a four-digit location code; a two-digit code designating the rack and panel number; and a two digit x-y rack coordinate. Token-Ring components are identified by a ten-digit number. The first five digits are the same as the cabling scheme. Digits six through nine indicate the unit number; the last two digits identify the receptacle.

Wire installation will probably be the most time-consuming task in the process. Factors affecting installation time and cost include: accessibility of the work area and path back to the wiring center; whether the wire installation needs to be done by Union personnel;

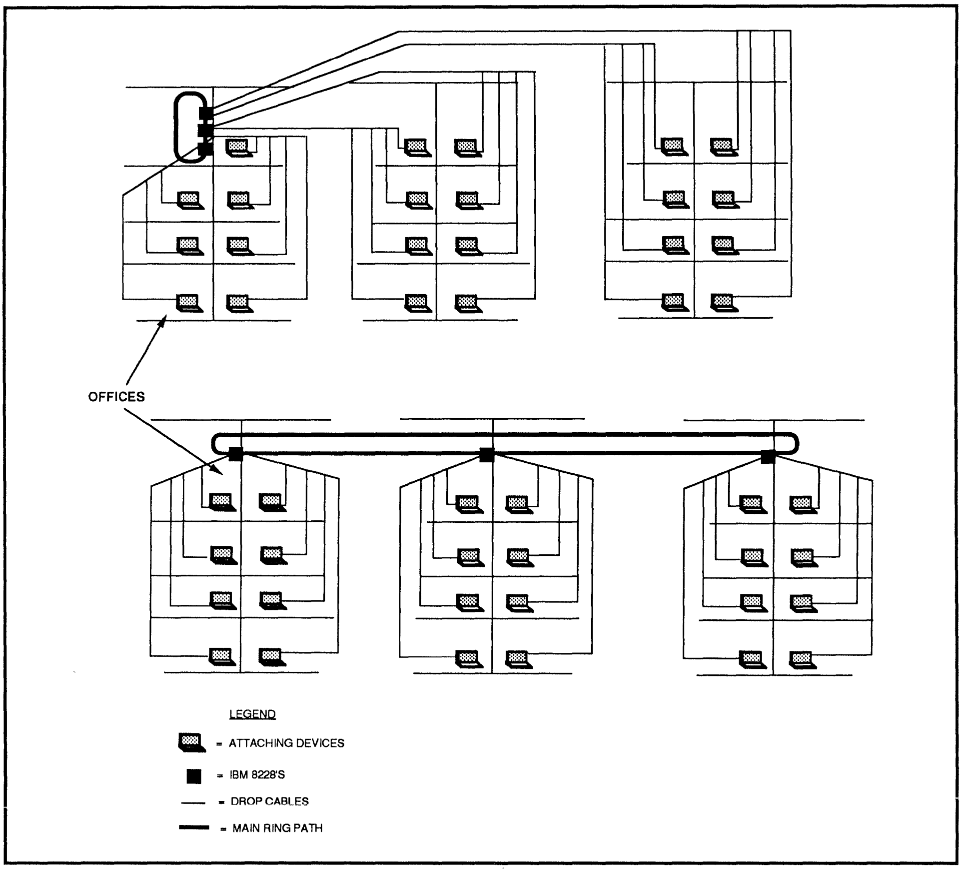

Figure 7-2: Cable Layout

and the time spent stripping wires, crimping connectors, and installing faceplates, etc.

Once the cable is installed, testing is performed before any devices are connected to it. IBM sells a cabling system tester which detects faults in copper data and telephone wiring by measuring continuity in the cabling system.

Continuity is described as an uninterrupted data or telephone conductor with a resistance of less than 500 ohms. An open in the cable is a data or telephone conductor shield which is normally unconnected and has a resistance greater than 10,000 ohms. A break is a normally continuous conductor/shield which has been interrupted and has a

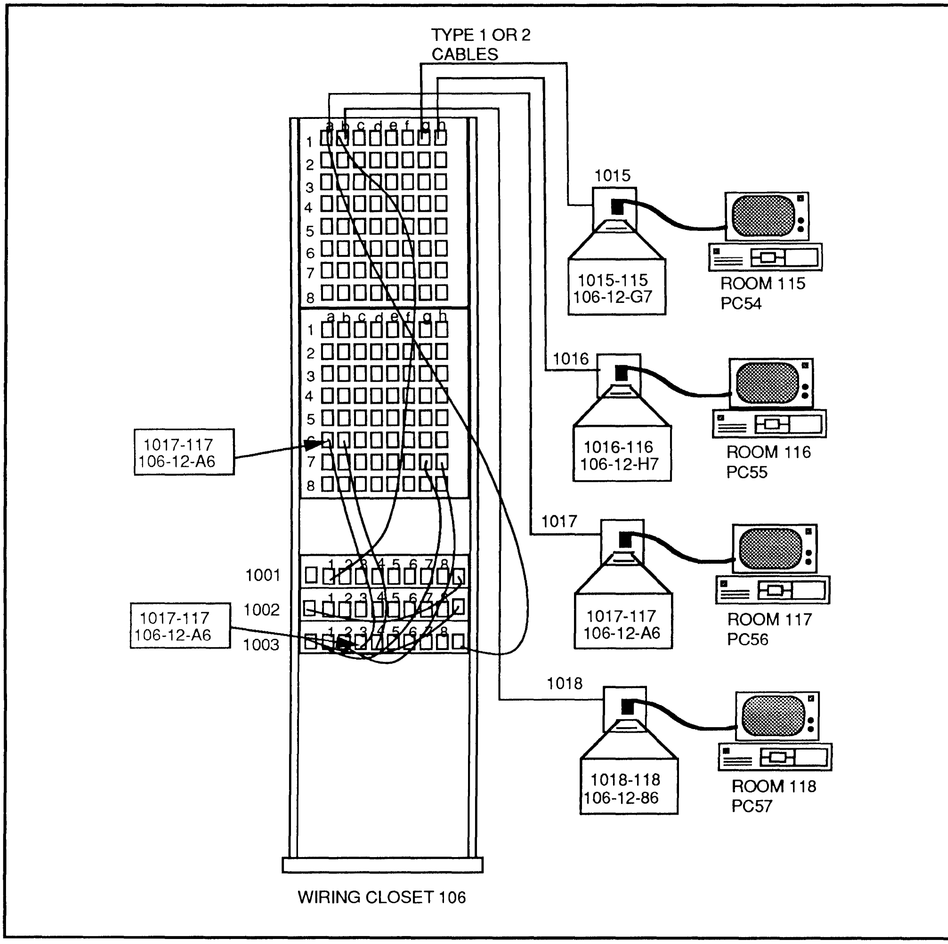

Figure 7-3: Component Labeling

resistance greater than 500 ohms. A short circuit describes a connection of two normally unconnected conductors shields with a resistance less than 1000 ohms. The IBM cable tester is a hand-held instrument using red and green LEDs to indicate any of a number of anomalies.

For fiber optics (Type 5 cable), a separate tester is available.

Cabling System Planner

The Cabling System Planner from Architecture Technology Corporation, is designed to assist with the planning and administration of an

IBM Cabling System. The Planner allows one to layout, in grid fashion, various cabling system components and define the cable necessary to implement a token-ring LAN. (To ensure accurate placement and sizing, a blueprint of the floor plan may be consulted.) Once the configuration is defined, the Planner can be used to change the layout using functions to re-define, move, and delete components. Equipment with an address assigned to it (such as a PC Adapter or TIC in a 3725) can also be associated with a faceplate.

Components are placed on an X-by-Y grid by defining a descriptor for that component. The descriptor contains detailed information about the component, such as its name and type of cable connected to it. A special descriptor known as a wiring closets contains details of multiple components.

Components that can be placed on the grid include faceplates, Multistation Access Units, repeaters, and surge suppressors. Components are labeled according to IBM's recommendations for its cabling system. The cable lengths can be computed in one of two computer methods or entered manually.

Multiple layouts are supported, allowing one to define several floors, multiple rings, more than one building, etc. Multiple rings may also be defined on the same layout. Once the layout is defined, several reports can be generated, such as locator charts, ring sequence charts, cable violations (for 4- and 16-Mbps rings), a summary, costs, ID labels, and complete component and cabling listings.

Consistent use of tools such as the Planner, enables one to track and manage the cabling system as it changes and grows in size. An alternative to the Planner is the IBM Cable Data Management System (CDMS) -- unfortunately, this program is hard to obtain since it was withdrawn from marketing. It had some interesting capabilities including the ability to generate plots of the cable layout. It may have been withdrawn due to its complex operation and not-so-friendly user interface.

Extending Rings

Rings may be extended in distance using repeaters (discussed in Chapter 2) and in distance/total stations using bridges.

Bridges may be implemented for several different reasons. The most common rationale for bridges is to increase the number of devices supported beyond the basic 260 node per ring limitation using data grade wire or the 72 node limitation using telephone grade twisted-pair. Using hierarchical and backbone techniques, literally thousands of devices can be supported. Careful attention to detail placement of bridges (by topology, ring load, and application usage) will ensure optimal performance.

Bridges can also be used to enhance performance and manageability. Performance can be improved by "isolating" groups of users by function (such as the programming department) and providing bridges for access to corporate- wide resources such as electronic mail. By partitioning the Token-Ring using bridges, it also becomes easier to manage in terms of problem determination, resource allocation, and security.

Token-Ring Network Bridge Program

The IBM Token-Ring Network Bridge Program operates in a PC AT or PS/2 with Micro Channel between two Token-Rings using a technique known as source routing (refer to Chapter 3 for a discussion of source routing, and Appendix C for other vendors of bridges). Up to seven levels of bridges are supported and they may operate in parallel for increased performance or to provide alternate routes for fault tolerant operation. A frame traverses the bridge in approximately 12 to 17 milliseconds.

The Bridge Program requires two Token-Ring Adapter IIs for the PC AT or two Token-Ring Adapter/As for the PS/2 Micro Channel. All workstation software that uses the 802.2 LLC interface and source routing, can operate transparently across the bridges. All IBM Token-Ring software including NETBIOS and APPC/PC meet these requirements. For non-IBM products, one needs to consult with the vendor of that product. As an example, it took Novell three years to make NetWare work across source-route bridges.

The IBM bridge program monitors error conditions and can give the location and time of occurrence of errors for either of the two attached rings. Beginning with the Bridge Program Version 1.1, various error and statistic information can be sent to the IBM LAN Manager Program. The Ring Parameter Server (RPS) provides the ring number to stations as they insert into the ring, and notifies the LAN Manager of the insertion. The Ring Error Monitor (REM) compiles error statistics. The Configuration Report Server (RES) forwards configuration change notifications to the LAN Manager and allows the LAN Manager to query adapters as required and set station parameters for optimal ring operation. Another interesting feature is the ability to trace a specially marked frame through the network for connectivity analysis.

IBM Token-Ring Network Bridge Program Version 2.0 connects both 4-Mbps and 16-Mbps Token-Ring Networks in any combination. IBM Token-Ring Network Bridge Program Version 2.1 bridges remote Token-Ring Networks via leased lines. This program enables the creation of a single logical network composed of many 4-Mbps and 16-Mbps Token-Rings, whether local or remote. As with Version 1.1, Versions 2.0 and 2.1 provide network management support to the IBM LAN Manager 2.0 by forwarding ring and bridge error information.

Version 2.0 can be configured locally or by IBM LAN Manager Version 2.0 to communicate with other active bridges in the network to automatically configure the network single-route broadcast path. If this capability is desired, all bridges in the network must be configured to participate in the protocols. Because the existing versions of the bridge program do not support automatic configuration, they must be upgraded to Version 2.0 to utilize this new function.

Version 2.1 can be used to connect either local or remote rings. The local bridge configuration provides all of the functions and capabilities of IBM Token-Ring Network Bridge Program Version 2.0. The remote bridge configuration uses one PC or PS/2 with bridge software at each end of a point-to-point, leased, SDLC line. When the program is configured as a remote bridge, frames are transferred between two rings via a leased teleprocessing line operating in full duplex at speeds from 9.6 Kbps to 1.344-Mbps. The remote bridge can attach to the telecommunication network on one of the following ways:

- Via synchronous modems capable of providing the following interfaces at the indicated speeds:
  - EIA RS-232-C/CCITT V.24 at 9.6 Kbps to 19.2 Kbps
  - CCITT V.35 at 9.6 Kbps to 1.344-Mbps
  - X.21 bis/V.24 at 9.6 Kbps to 19.2 Kbps
  - X.21 bis/V.35 at 9.6 Kbps to 1.344-Mbps
  - X.21 (leased only) at 9.6 Kbps to 64 Kbps.

Via a multiplexer, such as the Integrated Digital Network Exchange (IDNX).

Via the T1D3 feature of the ROLM CBX (Models 8000, 9000, and 9751) using the CCITT V.35 interface and a CCITT V.35 to EIA RS-449/RS-422A Interface Converter.

The remote bridge configuration of IBM Token-Ring Network Bridge Program Version 2.1 provides the capability of filtering frames forwarded by the bridge. A programming interface will be specified that will allow a user program to determine whether frames are allowed to pass through the bridge. A sample filtering program is provided.

Because the performance of the remote bridge is gated by the TP line, network traffic flow should be considered when selecting the line speed. For example, the broadcast traffic generated by a large network could cause severe performance problems at the 9.6 Kbps TP line speed. Because the effective throughput at the slower line speeds is low, the network traffic flow is a determining factor in the number of concurrent links that can be supported by the remote bridge.

To compensate for timing delays created by the TP line, parameters such as frame size and protocol timer values may need to be adjusted in the application. Applications not allowing such adjustments may not be able to communicate across the remote bridge at the slower line speeds.

Token-Ring Network Manager

The IBM Token-Ring Network Manager Program is designed to assist the user in problem determination and error recovery for the Token-Ring. The program monitors the ring for failures and provides information and control functions to assist the user with network problem determination and recovery. The failures recorded by the program include errors associated with the network cabling system, access units, and attaching station network adapters. Up to 260 attaching stations on a single ring are supported.

Recorded errors include soft and hard errors. A soft error (such as a station insertion) is recorded when the error threshold for transient recoverable errors (e.g., intermittent) is exceeded. It can indicate the impending loss of a network resource. A hard error is recorded when a permanent error (such as a cut cable) causes the loss of a network resource.

Network errors are recorded by the program in an event log on a disk. Other network events, such as stations joining and leaving the network, are optionally recorded in the log. The recorded information can assist the user with network problem determination. Events logged during a specified time period and/or associated with a specified adapter (station) can be displayed and/or printed.

Certain network errors indicate a loss or impending loss of a network resource and will cause an alert message to be recorded in the alert file in addition to the event log. An alert condition is immediately signaled to the operator by an audible alarm and a highlighted indicator on the operator display. Messages from the alert file can be displayed by the network operator. Probable cause and the source causing the alert can be displayed. Such information can assist in isolating a failure to a specific component of the network.

Operator control functions are provided. A station adapter can be logically removed from the network to assist with problem isolation. Connectivity problems can be investigated by displaying all online stations or requesting a path test between two stations. Intensive recording mode (which causes all soft-error incidents to be recorded) and continuous network traffic generation are operator-controlled functions that are useful in isolating transient failures.

Station names can be assigned by the operator and stored in a name file. The program will recognize and display both the symbolic name and the hardware address for stations with an assigned name.

By using Network Manager Version 1.1 in conjunction with NetView/PC, IBM has strengthened what was a weak (read non-existent) management link between the Token-Ring and SNA hosts. As a NetView/PC application, Network Manager provides automatic alert forwarding to a NetView host via an SDLC link. In addition, an operator at a remote stand-alone NetView/PC can dial up, via an asynchronous link, to a Token-Ring-attached NetView/PC and monitor/operate all of the Network Manager functions.

Version 1.1 lives up to its claim of being what its name implies: a "Manager" program--Version 1.0 was more of a diagnostics program. Version 1.1 still retains features of 1.0 such as monitor the ring for hard (disruptive) and soft (intermittent) errors, log errors to disk and/or a printer, and identify fault domains.

The IBM Token-Ring Network Problem Determination Guide contains step-by-step problem determination procedures for the network, and is used with the Network Manager Program to isolate and resolve ring failures.

LAN Manager

The IBM LAN Manager Version 1.0 is designed to execute as an application under control of NetView/PC, or as a stand-alone program. It supports the IBM Token-Ring Network and the IBM PC Network

Broadband. The LAN Manager runs on a dedicated PC under PC-DOS 3.3.

In the IBM Token-Ring Network environment, the LAN Manager incorporates all the functions of the Token-Ring Network Manager Release 1.1, which provides problem determination and error recovery assistance for single-ring networks. It also provides alert-forwarding to a CNM (Communication Network Management) System 370 host through NetView/PC. The LAN Manager Program extends these capabilities to multi-ring networks (in conjunction with the IBM Token-Ring Network Bridge Program 1.1 or higher) and offers a single point of control for bridges and rings in the network.

The LAN Manager is designed to run under NetView/PC, or as a stand-alone program. It provides a set of functions to manage a station, using the IEEE 802.2 LLC protocol. These functions include problem determination, problem reporting, and critical device lost notification. As an application of Net-View/PC, alerts also can be sent to a CNM host.

Both the IBM PC 3270 Management (discussed further in this chapter) and LAN Manager Programs monitor any workstations which are capable of supporting the IBM 802 LLC, original TOKREUI and the PC LAN Support Program. This means that the Network Manager workstation can be configured with any IBM LAN adapter except the original IBM PC Network card.

IBM LAN Manager Version 2.0, a network management program, enhances LAN management capability by allowing management of mixed networks (IBM Token-Ring Networks and broadband IBM PC Networks interconnected by IBM local area network bridge programs) from a single LAN management station.

IBM LAN Manager Version 2.0, an OS/2 EE 1.1 programming application, allows management of mixed networks (IBM Token-Ring Networks and broadband IBM PC Networks) interconnected by IBM local area network bridge programs, and provides extensive monitoring, problem determination, and isolation capabilities for centrally managing the LAN from a NetView host or a station on the LAN.

LAN Manager Version 2.0 includes all appropriate LAN Manager Version 1.0 management functions. LAN Manager Version 2.0 supports Token-Ring Networks connected by the IBM Token-Ring Network Bridge Program Version 1.1, broadband PC Networks connected by the IBM PC Network Bridge Program Version 1.0, and mixed-media Token-Ring Networks and broadband PC Networks interconnected by the above bridge programs.

LAN Manager Version 2.0 operates as a LAN management agent for NetView running in the host or as a local LAN manager on the LAN. It exchanges alert and command information with NetView at the host via an OS/2 EE 1.1 Communications Manager-supported 3270 emulation session. The LAN Manager Version 2.0 can co-exist with NetView/PC Version 1.2 and uses its communication facilities.

Using NetView's Service Point Control Service (SPCS) facility to carry the appropriate parameters to the LAN Manager, NetView extends LAN management by allowing the NetView console operator to issue CLIST commands to, and receive responses from, the LAN Manager. Using CLIST commands, the NetView operator accesses 11 LAN management functions which request adapter, bridge, and network status information, and take action such as removing an adapter from the network.

With the Alert Transport Facility, the LAN Manager can receive and display generated alert frames from IBM Personal Computer DOS and OS/2 EE applications running in LAN-attached workstations and, as an agent of NetView, pass the alert message for display on the NetView operator console.

Critical resource monitoring, available for single-segment PC Networks in LAN Manager Version 1.0, is extended to Token-Ring Networks in LAN Manager Version 2.0. With this facility, selected LAN devices may be defined as critical resources. The LAN Manager monitors each selected device and sends an alert when the device becomes unavailable.

LAN Manager Version 2.0 can enhance trace security by maintaining a list of trace adapter addresses, verifying authorization, and removing unauthorized adapter addresses when used with the IBM Token-Ring Network Trace and Performance Program with IBM Token-Ring Network Trace and Performance PC Adapter II or IBM Token-Ring Network Trace and Performance Adapter/A.

LAN Manager Version 2.0 cannot use the IBM PC 3270 Emulation program for host communication. It communicates satisfactorily using the OS/2 EE 1.1 supported gateways. In environments where NetView/PC is required, LAN Manager Version 2.0 may be configured to run with NetView/PC.

3270 Emulation LAN Management

The IBM PC 3270 Emulation LAN Management Program Version 1.0 is a "poor mans NetView" in that it provides small and remote IBM Token-Ring Networks or PC Networks (with no LAN Manager Program) the capability for centralized network management. The program, residing under an IBM PC 3270 Emulation Program gateway, monitors the LAN for alerts and provides automatic alert-forwarding to a NetView host.

This program accumulates soft error information for the IBM Token-Ring Network, monitors the IBM Token-Ring Network for hard errors, monitors the IBM PC Network for hot carrier, no carrier, etc., and builds LAN-related alerts for transport to NetView.

The IBM 3270 PC Emulation Program is required to operate this program.

NetView

IBM intends to evolve network management toward enterprise-wide network management -- in other words, to incorporate the voice, data, and SNA functions of a corporate environment under one umbrella -- NetView. A typical network management environment today consists of customers who manage their own systems. One of the complexities in such an environment is that there are many views of systems -- and many views of the network: there are, for instance, data networks, voice networks, carrier networks, and multiple-vendor networks that people attempt to manage as a whole.

IBM believes that customer requirements include the ability to manage this multi-vendor environment (not just IBM equipment), pare down the network management cost, and reduce the complexity of managing a network -- in essence, IBM is just plain catching up on managing the exponential growth in recent years of terminals, personal computers, and workstations in general. The challenge here to IBM is to: provide an open-architected, enterprise-wide system solution; provide end-to-end management; and incorporate management for voice, data, imaging, graphics, and text. IBM wants to have automated and continuous operation, it wants to be able to concentrate on centralized management with options for distributed control, and, by all means, it wants to allow the customer to control this environment.

The IBM view of network management is shown in Figure 7-4. This IBM conceptionalization consists of a triangle with three anchors: one anchor is a distributed point of control for SNA resources, also known as an "entry point;" a second anchor is a distributed point of control for non-SNA resources, also known as a "service point;" and

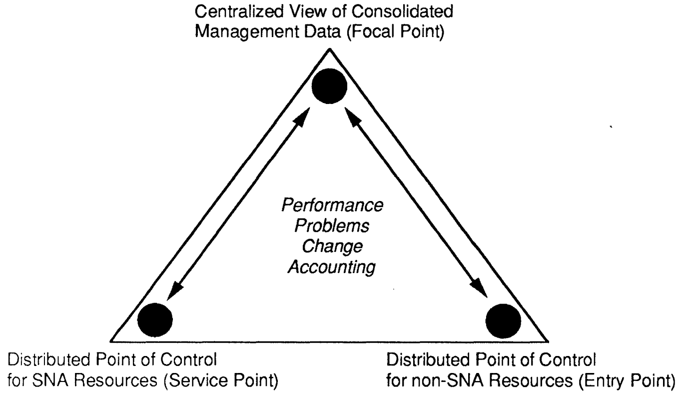

Figure 7-4: The IBM View of Network Management

both report to the third anchor, which is a centralized view consisting of consolidated management data, also known as a "focal point."

An entry point supplies general connectivity for the attached components to access end-user services provided by the network. An entry point is an SNA- addressable product that exchanges formatted network management messages, such as alerts and response-time monitor requests and responses, between its attached devices and a focal point.

A service point provides a connection through which network management data can be converted to SNA formats and transmitted to the focal point for processing. The service point is SNA-addressable and can accept SNA-formatted network management requests from the focal point, convert them, send them to an attached network component, and then send the component's response back to the focal point. Examples of IBM products that are service points are: NetView/PC; IBM LAN Manager Version 1.0; NetView/PC ROLM Alert Monitor; and NetView/PC ROLM Call Detail Collector.

The focal point manages all remotely and locally attached network components with respect to one or more network management disciplines (for example, problem management, change management, or operations control).

NetView is an integrated network management product that consists of installation aids, a command facility, status monitor, a help disk, and browsing capabilities. The command facility within NetView is a base for centralized network management that includes: session monitoring to monitor response time; session configuration; problem determination; data collection; suggested remedial actions when problems or alerts occur; and a status monitor that provides for network status display and automatic reactivation.

NetView/PC provides local capability in terms of management, processing, services, and operations, with the ability to report to a remote NetView that runs on a host. This concept is illustrated in Figure 7-5.

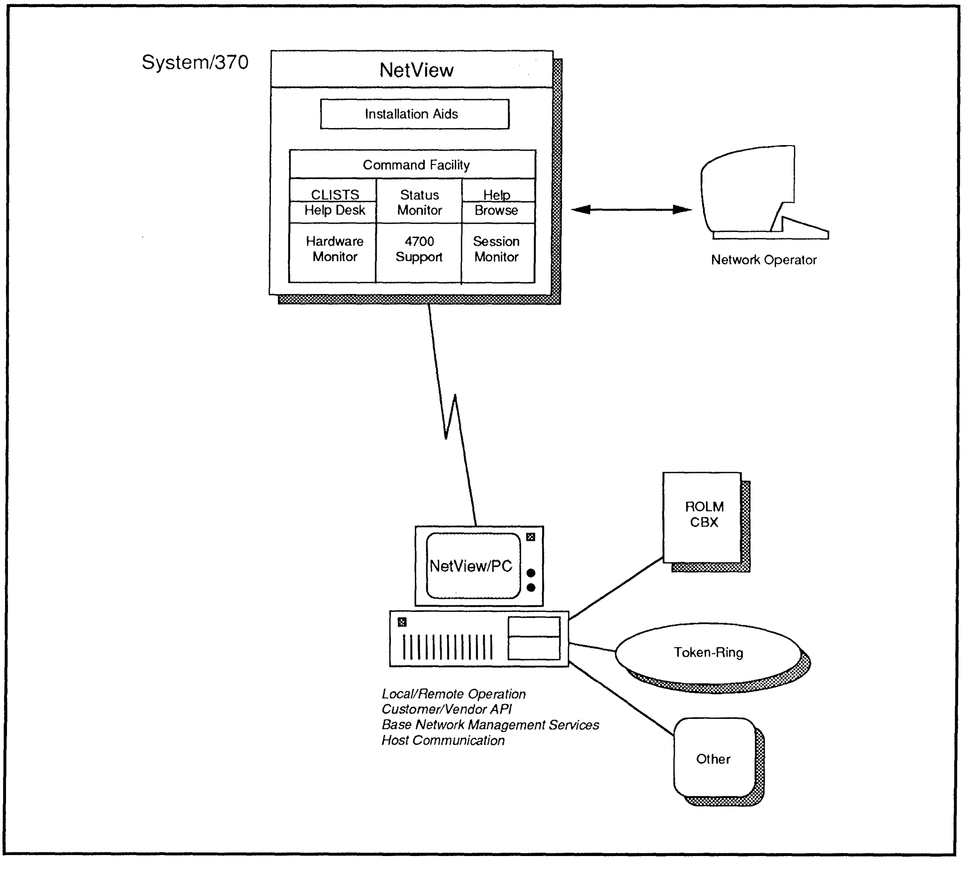

Figure 7-5: NetView/PC

Operations management functions presently provided by NetView include remote operation capability from System/370 or System/36; message automation; and centralized control of Service-Point-supported devices. Future requirements for operations management with NetView include the ability to more directly monitor the hardware; enhance automation and ease-of-use; and provide continued product integration.

Currently, change management functions supported by NetView include planning, distribution, installation, and tracking. Future requirements include a non-disruptive change for unattended equipment, support for customization of the software, the distribution of NetView from a central, common database, and a configuration manager.

Configuration management includes a configuration description capability and a temporary dynamic configuration update. Future requirements in this area include a reduced system definition, non-disruptive definition, dynamic configuration update (making it permanent instead of temporary), a common data repository, and access to shared data.

Problem management includes alert capability (from various SNA components, the ROLM CBX, and the Token-Ring Network Manager or LAN Manager), diagnostics capability, problem tracking, and a service reminder. Anticipated directions in this area of problem management include continuous problem determination, non-disruptive switching, and enhanced access to problem management data.

NetView/PC, then, is a multitasking personal computer subsystem and an implementation of the service point in IBM's Open Communication Architectures. It is the base upon which separately announced, device-dependent CNM applications provide support for the IBM Token-Ring Network Manager Version 1.1 and products/capabilities support for the ROLM CBX by NetView/PC ROLM Alert Monitor and NetView/PC ROLM Call Detail Collector. The NetView ROLM Call Detail Collector also supports other selected PBXs.

NetView/PC provides the base services required by a device-dependent CNM application program and a network technician/operator. The key services can be grouped into the following three categories: (1) services available to IBM or ROLM device-dependent applications -- file manager, communications manager, multitasking facilities; panel, display, keyboard facilities; (2) services available to a user-written IBM PC-DOS program -- host file transfer and communication with NetView; and (3) services available to a technician/operator -- operator control, general help, remote console; problem management, alert management, service reminder.

NetView/PC's multitasking support enables its facilities -- such as the Alert Manager, the Problem Manager, and others -- to be concurrently active. The user, through the use of a "hot key," can easily switch from facility to facility.

The Alert Manager Facility stores alert data captured by IBM or ROLM-developed or user-written applications. It permits an operator to display and/or delete the captured data. The Alert Manager provides the option to create a problem record associated with the alert data. When NetView/PC is host-connected, the alerts are automatically sent to NetView.

The Problem Manager facility permits a user to create, delete, or modify network problem information. The Problem Manager, in association with the Service Reminder function, allows tracking and managing of network problems.

The Service Reminder facility is used as an "alarm clock" to remind NetView/PC operators of the need for such things as network configuration changes, completion of service work, or any other planned task.

The DOS Partition Support permits the NetView/PC operator to execute PC-DOS commands or a single, well-behaved user-written application while NetView/PC is active and operating.

Data captured and logged by IBM or ROLM-developed applications is converted into data interchange format (DIF). This data can be processed by report generation programs such as the REPORTS+ program of the IBM Personal Decision Series.

The Remote Console facility permits NetView/PC operating on one IBM Personal Computer to be controlled remotely through an ASCII/asynchronous telecommunications link from another IBM PC running NetView/PC.

The Communications Manager consists of application program-to-application program communication services (LU6.2), SNA physical unit services, and the services required to communicate to remote devices over an RS-232-C interface using either synchronous data link control (SDLC) or asynchronous/ASCII protocols. This facility provides the communications services to IBM or ROLM device-dependent applications and the remote console facility of NetView/PC. Most of the communications manager components are resident in the IBM Realtime Interface Coprocessor's memory.

The LU6.2 service component is used to provide file transfer capability between NetView/PC and the host CICS/DDM licensed program. The SNA physical unit services component is responsible for managing the NetView/PC end of the System Services Control Point (SSCP)-to-physical unit session with the owning ACF/VTAM. The SSCP-to-physical unit session is used to transmit alerts to NetView and for exchanging CNM data between NetView and NetView/PC.

API/CS is an interface between device-dependent applications and NetView/PC. It allows user-written assembler language applications to use a subset of the NetView/PC's base services. The applications run in the DOS Partition and are written in IBM Personal Computer Macro Assembler Language. Standard DOS and IBM Personal Computer BIOS function calls are available to the application programmer. The following services are available to the application: send an application-captured alert to NetView; initiate a file transfer between NetView/PC and CICS/DDM; and exchange CNM data with a user-written NetView application program.

NetView/PC Version 1.2 operates with OS/2 and takes advantage of the memory protection and increase of usable memory provided by OS/2. Other Version 1.2 enhancements apply in the areas of Open Applications Program Interface (API) for asynchronous communication, C language sample program, local display of generic alerts, simplified generic alerts generation, Service Point Command Facility (SPCF), alternate lower-cost host connection options, and Japanese translation.

The migration of NetView/PC from the PC-DOS to IBM OS/2 Extended Edition Version 1.1 base provides the following enhancements:

- Memory protection for NetView/PC applications. This provides protection for NetView/PC application programs from other application programs if defective software is encountered.
- Increase of usable memory. OS/2 allows addressing up to 16 Mbytes of memory. This will allow more concurrent NetView/PC applications. Also, multiple NetView/PC vendor

and end-user applications may use operating system functions concurrently.

The API/CS is extended to provide an asynchronous communication service API for vendor and end-user applications and allow application use of the Realtime Interface Co-Processor. The commands available are OPEN, SEND, RECEIVE, and CLOSE. This also permits vendor and end-user applications to use the IBM Realtime Interface Co-Processors independently or concurrently with other NetView/PC applications.

Local generic alerts generated by NetView/PC applications and sent to the host can be viewed at the NetView/PC workstation. Previously, these alerts could only be viewed at the NetView host terminal.

Facilities are provided in the API to allow an application developer to build and modify generic alerts more easily. This is similar to the facilities already provided with the Service Point Command Facility (SPCF).

Chaining of SPCF responses is provided to allow applications to reply to an SPCF command with more than 512 bytes of data. Applications can send a greater quantity of data to the host with a single request. SPCF performance may also be improved.

The number of applications that can connect to SPCF at any one time has been increased from one to 128. This allows multiple applications to communicate with SPCF simultaneously.

SPCF will notify the application of the receipt of SPCF commands and responses intended for the application. The application no longer polls the SPCF, and application performance may also be improved.

The Alternative Host Connection function allows host connection without requiring the IBM Realtime Interface Co-Processor card. Commands, alerts, and host data file transfer will flow on the host sessions through the OS/2 Extended Edition Communications Manager.

NetView/PC Version 1.2 communicates with the host-based program NetView to send alerts and to exchange other CNM data. Non-generic alert information, recommended actions, and probable cause information for LAN and CBX are predefined in NetView Release 3 and are available to the network operator.

Data files can be sent between NetView/PC and a host computer running the CICS/DDM licensed program operational in the host computer.

NetView/PC Version 1.2 is not upwardly compatible with NetView/PC Version 1.1. NetView/PC applications must be re-compiled and/or re-linked. Additional conversions may be required depending on the application interface with the operating system. Due to differences between the NetView/PC Version 1.1 Operating System (IBM PC/DOS) and the IBM OS/2, the same version of NetView/PC is required on both the local and the remote consoles.

NetView/PC Version 1.2 supports NetView Release 1, NetView Release 2, or NetView Release 3. The earlier NetView releases, however, may not be the same levels of the SPCF functions supported as NetView Release 3 when interacting with NetView/PC Version 1.2.

The LAN Manager Version 2.0 will operate on NetView/PC Version 1.2.

IBM intends to provide additional support in the future for the following:

<u>Remote Console Support On OS/2 Extended Edition</u>: The NetView/PC Remote Console facility will be converted to run on the OS/2 Extended Edition. The Remote Console facility permits NetView/PC operating on one IBM workstation to be controlled remotely through a telecommunications link from another IBM workstation running NetView/PC.

<u>Asynchronous API Via The OS/2 Extended Edition Communications Manager</u>: This function will extend the open API provided in Version 1.2 for asynchronous communications through the OS/2 Extended Edition Communications Manager. An application program can use the open API for asynchronous communications and use either the IBM Realtime Interface Co-Processor or the OS/2 Extended Edition facilities. This allows the Realtime Interface Co-Processor adapter to be optional. The Realtime Interface Co-Processor adapter option should still be considered for these environments where the number of communication ports and speed are important parameters.

NetView is clearly IBM's strategic network management tool. IBM has spent a good deal of time changing and refining this tool. Early reports from NetView users on S/370 machines indicate that it eats up many processing cycles. Another problem is that in large SNA networks, information collected by NetView can quickly use up large amounts of storage. So future enhancements to NetView will need to include addition data filtering capabilities, as well as incorporation of yet more IBM network management packages such as Network Management Performance Monitor (NMPM) and Operator Communications Control Facility (OCCF).

Overall, to support this network management direction, IBM must provide an open architecture by: publishing the architectures; publishing the interface to the API; and participating in network management standards committees.

Bridging to Ethernet

There are many hurdles that must be overcome when trying to bridge the token-ring to Ethernet. In fact, it took four years (since the token-ring was announced) for vendors to beginning shipping the first token-ring to Ethernet bridges.

One of the most difficult and subtle hurdle has to do with the fact that all token-rings operate 802.2 LLC (since IBM bundled it into the token-ring adapters). All Ethernets do not. In fact, the majority of the Ethernets evolved from the Xerox standard (Ethernet was the basis for 802.3), which specified the Xerox DLC (data link control) for link control, not 802.2 LLC (DLC was actually developed before LLC became a standard).

Another problem is that IBM has introduced a routing mechanism at a very low level in the system -- at the 802.5 MAC level or layer 1 and 1/2. Recall from Chapter one that ISO protocols do routing at layer 3.

Yet another problem is that some of the bit ordering (the order in which bits are transmitted and received on the media) on Ethernet is different than token-ring.

This brings up an interesting point -- despite vendor claims, MAC-level bridges are not protocol independent! They must deal with these layer one differences. And once that is resolved, a DLC/LLC translation/emulation must be done -- otherwise the data is totally useless above layer 1! Once these issues are dealt with, it is up to the high level protocols to be compatible (this is another problem in itself, beyond the scope of this report) on both sides before we can have transparent operation between the two networks.

The following describes the bit ordering problem in more detail.

The transmission of data for 802.3 and 802.4 LAN mediums occurs least significant bit (LSB) first. This is true for the entire packet: LAN MAC address fields (source and destination), MAC specific fields (e.g., length field in 802.3 LANs), and the MAC information field.

On an 802.5 LAN medium, the LAN MAC address fields (source and destination) are transmitted such that the first bit on the medium is the Individual/Group Address bit group, in a fashion similar to that of 802.3 and 802.4 LAN media; the MAC information field, however, is transmitted most significant bit (MSB) first on an 802.5 LAN medium. The MAC information field is defined to be that part of the frame starting directly after the MAC header and including all the data up to, but not including, the frame check sequence (e.g., LLC header information, such as the protocol identifier field, is contained in the MAC information field).

For frames which originate within the MAC (e.g., MAC embedded management frames), the ordering of bits within the MAC information field is defined by the 802.5 MAC specification.

The follow describes a product from IBM, which resolves some of these incompatibilities between token-ring and Ethernet.

IBM 8209 Bridge

In September of 1989, IBM surprised the LAN marketplace by announcing a bridge for connecting an IBM Token-Ring LAN to Ethernet. The new bridge accommodates Ethernets without IEEE 802.2 LLC protocols (the majority of Ethernets), as well as Ethernets that do implement LLC (such as those from Ungermann-Bass). The product will initially be able to recognize TCP/IP frames and pass them through while blocking non-TCP/IP frames. Source routing continues to be supported on the Token-Ring side, while the 802.1 spanning tree algorithm is implemented on the Ethernet side.

This was the first product to ship which supports both routing protocols (since then, CrossCom has also shipped such a product). Obviously, IBM feels that the "source route" vs. "spanning tree" debate is not an issue and will continue to pursue source routing as a standard for 802.5. As it turns out, some of the spanning tree algorithm is already used in IBM's token-ring bridge product for single-route (limited broadcast) functions.

The IBM 8209 LAN Bridge interconnects an IBM Token-Ring Network with an Ethernet Version 2 or IEEE 802.3 LAN. The 8209 handles the conversion necessary to route information between the dissimilar LANs. IBM and OEM systems and workstations using this connection require compatible protocols such as TCP/IP, OSI, SNA, NETBIOS, or IEEE 802.2 in order to communicate.

The 8209 bridges the LAN physical layer differences by providing two LAN ports, one Token-Ring Network and the other Ethernet/IEEE 802.3. The Token-Ring port operates at either 4 or 16-Mbps. When the 4-Mbps rate is selected, the Early Token Release function will be disabled; when the 16-Mbps rate is selected, Early Token Release will be enabled, unless specifically disabled by the customer using the 8209 Utility program.

The Ethernet/IEEE 802.3 port attaches to either an Ethernet Version 2 or IEEE 802.3 LAN. The 8209 accommodates these two CSMA/CD LANs through two modes of operation; the mode of operation is determined by configuration switches. With the switches set for Automatic Mode Detection, the 8209 will examine the data stream and dynamically adapt to the correct operational mode. This allows both modes to be used simultaneously. However, if the destination address of the Ethernet/IEEE 802.3 attached station is not in the 8209 database, the correct format cannot be dynamically determined. In this case the 8209 will perform format conversion based on the setting of the "Mode 1/Mode 2 Priority Operation" hardware configuration switch.

Mode 1 is used to bridge Token-Ring to Ethernet Version 2 LANs. Above the physical layer, Token-Ring and Ethernet Version 2 differ in Media Access Control (MAC) and user Datagram services. The Token-Ring MAC layer complies with IEEE 802.5, while Ethernet uses Carrier Sense Multiple Access/Collision Detect (CSMA/CD) and does not specify an 802.2 Logical Link Control (LLC) Interface. Additionally, Token-Ring offers both Type 1 (connectionless) and Type 2 (connection-oriented) LLC services; there is no Ethernet Version 2 LLC equivalent function. The 8209 LAN Bridge handles this difference in Mode 1.

LLC information frames, originating on Token-Ring and destined for stations on Ethernet, are stripped of the LLC protocol, converted to Ethernet Version 2 frames, and transmitted on the Ethernet port. A slightly more complicated process is used to convert and forward Ethernet frames onto the Token-Ring. The 8209 retrieves Token-

Ring routing information from its data base and inserts it in the frame along with a Sub-Network Access Protocol (SNAP) header.

TCP/IP exchanges between stations on the different LANs would use operational Mode 1.

Mode 2 interconnects Token-Ring and IEEE 802.3 LANs. This mode provides IEEE 802.5 (Token-Ring)/IEEE 802.3 MAC-level frame conversion. Protocol layers above the MAC are transparent to the 8209 LAN Bridge and are passed through without modification.

SNA, NETBIOS, OSI, or IEEE 802.2 sessions between stations on the different LANs would use the bridge in Mode 2.

The 8209 LAN Bridge transparently handles conversions for routing information between the two LANs, regardless of the operational mode (Mode 1 or 2). To stations on the Token-Ring Network, the 8209 appears as a bridge to another Token-Ring. A station on Token-Ring uses source routing to communicate with any station on Ethernet/IEEE 802.3. The 8209 is functionally transparent to stations on Ethernet/IEEE 802.3, appearing as one or more native Ethernet/IEEE 802.3 stations. Ethernet/IEEE 802.3 attached stations can communicate with Token-Ring attached stations as if they were on the same LAN.

The 8209 LAN Bridge maintains two data bases: one contains station addresses for Ethernet/IEEE 802.3 stations; the other contains Token-Ring station addresses and routing information. The Ethernet/IEEE 802.3 data base consists of static and dynamic entries: the Token-Ring data base, however, contains only dynamic entries. Static entries differ from dynamic entries in that they are configured, loaded, and retained in a non-volatile RAM. Dynamic entries are created as part of the 8209's "learning process" and are lost when power is removed.

After power-on initialization, the Ethernet/IEEE 802.3 data base is initialized with static entries. The 8209 enters the learning state, listens to all frames on the Ethernet/IEEE 802.3, and saves each unique source address in the data base. While in this state, the 8209 will not forward any frames. After a few seconds the bridge leaves the learning state and begins normal operations. During normal operations, the 8209 updates the data base when a new source address is detected on an Ethernet/IEEE 802.3 frame.

The Token-Ring data base is dynamically built during normal operations. Entries will be added to the Token-Ring data base only for stations with frames that are forwarded on the Ethernet/IEEE 802.3 port.

The 8209 provides for a combined total of 2048 data base entries. The Token-Ring data base portion of this total may be from 1 to 1024; the number of entries in the Ethernet/IEEE 802.3 data base, however, may range from 1 to 2047. An Aging Timer will determine how long inactive Dynamic Entries will remain in the data base. Static data base entries are not subject to removal by the Aging Timer.

Configurable filters are employed to reduce forwarding of unnecessary traffic. For example, filters may be set up so that only TCP/IP traffic is forwarded to either port. This means even though general broadcast frames are received from either port, they will be discarded if not identified as TCP/IP. Filtering improves performance for the 8209 and ensures no degradation of the attached LANs occurs due to unnecessary traffic.

The 8209 ensures that no traffic is held in the bridge longer than the maximum transit time. If, due to network congestion, the 8209 is unable to forward traffic and the traffic is held longer than permitted, it will be discarded. This function is useful for those high-level protocols that generally retransmit a message unless an acknowledgement is received within a certain time.

Each 8209 has a unique MAC address assigned for each port. The IEEE universal address administration process will be used to assign the MAC addresses.

The 8209 provides Token-Ring Network LAN Reporting Mechanism, Ring Parameter Server, and LAN Bridge Server functions (as defined in the *IBM Token-Ring Network Architecture Reference*). The 8209, acting as a logical LAN management agent on the Token-Ring Network, will:

- Keep and report statistical frame traffic information
- Accept and respond to requests for Bridge and Route status
- Accept and respond to commands which change Bridge configuration parameters

The controlling LAN Manager can set/reset these configuration parameters:

- Notification interval for performance statistics
- Bridge internal status

- Route active status
- Hop Count
- Ring number
- Bridge number
- Enabled functional addresses
- Reporting functional addresses

The LAN Manager will be installed on a Token-Ring station and will establish a link to the 8209 LAN Bridge in order to communicate these management functions. The 8209 can provide management information to the LAN Manager only for the Ethernet/IEEE 802.3 segment to which it is attached.

A utility program is shipped with the 8209. This utility runs under either PC DOS or OS/2 in an IBM PC or PS/2. The utility allows the customer to set up filters, static data base entries, ring numbers, bridge number, and timers which control operation of the 8209 LAN Bridge. The IBM PC or PS/2 communicates the configuration parameters to the 8209 over the Token-Ring. The 8209 Utility program allows the customer to examine and modify the following parameters:

- Spanning Tree Parameters
- Operational Mode
- Enable/Disable Early Token Release
- Filter Definitions
- Ethernet/IEEE 802.3 Static Database Entries
- Ethernet/IEEE 802.3 Port Statistics
- Bridge Number
- Token-Ring LAN Number
- Ethernet/IEEE 802.3 LAN number

In addition to configuration, the 8209 Utility program provides a means to collect Ethernet/IEEE 802.3 port statistics which are gathered by the 8209.

IBM was clever in making the 8209 look like its existing Token-Ring bridges, complete with the network management functions. Another interesting observation is that one can now construct scenarios with Ethernet operating as a backbone for Token-Ring networks!

IBM customers will probably like this product. However, marketing into non-IBM accounts could be a problem. Confidence in the 8209 for these customers won't be there until IBM can prove interoperability with existing spanning tree products and Ethernets operating the popular TCP/IP protocol.

Protocol Analyzers

In addition to checking for problems and performing protocol analysis, LAN protocol analyzers are extremely useful when debugging your own network protocols or when developing systems software that rely on the underlying vendor's existing protocols. Those elusive bits and overdue acknowledgements can be "sniffed out."

One such useful tool appropriately named "The Sniffer" that was designed by Network General Corporation in Menlo Park, CA. The Sniffer joins the fray of LAN protocol analyzers from vendors such Vance, Excelan, and HP. What makes The Sniffer unique is that it can operate on not only Ethernet, but was first developed for the token-ring. The Sniffer is offered with a suite of protocol interpreters that can be used to break down protocols, layer-by-layer, vendor-by-vendor.

The Sniffer is offered with 80286- or 80386-based portable PCs, adapter(s), Sniffer software, and DOS. The machine can be ordered with an Ethernet adapter (3Com Etherlink Plus), a token-ring adapter (Proteon token-ring AT), or both.

It is interesting to note that the Token-Ring was designed to be more secure than Ethernet by leaving out a promiscuous mode option where all frames can received by any station. For this reason, the Proteon adapter uses a modified TI chip-set that is normally used only in the initial design of a token-ring adapter.

The available protocol suites include: several TCP/IP varieties (including ARP, TCP, UDP, ICMP, DNS, Telnet); Sun Microsystems NFS protocol (including RPC, YP, and PMAP); ISO 8473 IP (internet); ISO 8073 TP class 4 (transport); XNS protocols used by Xerox, 3Com 3Plus, and Ungermann-Bass (including PEP, SPP, RIP, and Courier); the Server Message Block (SMB) protocol developed by Microsoft for MS-Networks and also used in the IBM PC LAN Program; NETBIOS; NetWare Core Protocol (used for dialog between the workstation and a file server); the IEEE 802.2 logical link control (LLC) Type 1 and Type 2 protocols (used by virtually all token-ring and a few Ethernet LANs); and the 802.5 media access control (MAC) protocol.

One can add custom protocols or write their own interpreter for protocols supported by Network General, such as TCP/IP. More importantly, one can generate several versions of the EXE file for different networks.

The basic operation of The Sniffer is that it collects all packets and breaks them down into higher-level protocols based on the frame format. Packets can be filtered according to a single station address, a particular protocol such as NetWare identified packets (using the service access point or SAP) at the IEEE 802.2 LLC level (LLC is the highest level that may be filtered), or a pattern match (limited to 4 bytes plus an offset).

Once packets are in the capture buffer (up to 8 Mbytes of RAM), they may be displayed or saved for later analysis. This buffer can fill up rapidly even with filters set (the user can decide whether or not to capture in a circular buffer fashion that overwrites previously captured frames or to stop when the buffer fills). The only way to save the buffer is to momentarily stop the capture, save to a file, and start a new capture (that erases the previous buffer). Some protocol analyzers such as Excelan's Lanalyzer can pipeline the data to a hard disk file as it is received.

End users never see the underlying protocols that carry their data safely and swiftly the network but network designers have nightmares about them! Protocols have very precise syntax and semantics associated with them. Many vendors will tweak and modify them making them incompatible with other vendors. For example, this is what Novell did when it used the basic XNS Internet packet format as the basis for the high level NetWare protocols.

"Sniffing" the token insertion sequence is a good way to examine the operation of MAC frames. This sequence is illustrated in Figure 7-6, which is a Sniffer generated print out of the frames in summary format. (The printout shows only the high-level protocol within in frame, since it was selected for viewing. Whatever you select for viewing (summary, detail, or hex) can also be printed or saved to a DOS file.)

The columns are as follows: frame number, relative time from beginning of capture (in milliseconds -- the resolution of The Sniffer's time-stamping), the destination name, the source name, and the summary of the frame protocol. The source and destination can be displayed as either the 48-bit token-ring address or a name, since the user has the capability to associate names with addresses for easier station recognition. The time can be displayed/printed three different ways: by relative time, absolute (real-time clock) time, or by the delta since the last frame. The delta since the last frame was somewhat peculiar. Sometimes it displayed 0.000 delta when the relative time showed that there was .001 between certain frames. Apparently there is a rounding error of some sort in the delta time algorithm.

Before inserting JSH AT into the ring, from Figure 7-6 we can see The Sniffer is the active monitor while the NetWare server is the standby monitor. The monitor MAC frames are broadcast very close to every seven seconds. Frame 7, at 18.762 seconds is a Ring Purge, by the active monitor meaning that the token was lost --the likely cause of the disruption in this case is that JSH AT has activated the relay in the Multistation Access Unit (MAU), interrupting the normal flow, causing the lost-token-timer to expire at the active monitor and thus a ring purge to be issued.

After the ring purge, the token is flowing again and in frames 8 and 10, JSH AT checks to see if any other station has its address. No other stations do so the next step is to notify the LAN Manager (this is an IBM unique management function) that there has been a change in the upstream address of JSH AT. The LAN manager is targeted by broadcasting a functional address (a special broadcast group address) that can be picked up by any station listening for that address. In our testbed, there is no LAN Manager, so this frame goes unprocessed. In frame 12, it was time for JSH AT to broadcast its standby monitor frame.

In frames 14 to 17, we see a burst of Request Initialization frames that are sent to the functional address for a parameter server -- also lacking in our networking. The IBM Token-Ring Bridge Program, however, also acts as a parameter server and supplies information such as a ring number (for use in source routing as well as problem determination). From this point on, JSH AT is ready to use the Token-Ring to communicate with NetWare or any other resource on the ring.

In addition to carrying MAC information, the basic token-ring frame (or 802.3 Ethernet for that matter) may also carry LLC frames. An important LLC field is the destination service access point (DSAP) which tells the frame handler at the receiving side which process to pass the frame off to. For example, a DSAP of E0 hex will be passed off to resident NetWare software (such as the shell or the file server) whereas a F0 hex will be passed off to a NETBIOS handler.

```
Sniffer data collected on 5/22/87 at 14:57:36, file C:\CAPTURE\ATC_TR\3H.TRC, Page 1

SUMMRY  Delta T     Destination    Source       Summary
M  1                Broadcast      Sniffer      MAC Active Monitor Present
   2    0.013       Broadcast      NetWare      MAC Standby Monitor Present
   3    6.913       Broadcast      Sniffer      MAC Active Monitor Present
   4    0.014       Broadcast      NetWare      MAC Standby Monitor Present
   5    6.912       Broadcast      Sniffer      MAC Active Monitor Present
   6    0.015       Broadcast      NetWare      MAC Standby Monitor Present
   7    4.892       Broadcast      Sniffer      MAC Ring Purge
   8    0.000       JSH AT         JSH AT       MAC Duplicate Address Test
   9    0.000       Broadcast      Sniffer      MAC Active Monitor Present
  10    0.000       JSH AT         JSH AT       MAC Duplicate Address Test
  11    0.000       LAN Manager    JSH AT       MAC Report SUA Change
  12    0.013       Broadcast      JSH AT       MAC Standby Monitor Present
  13    0.000       LAN Manager    NetWare      MAC Report SUA Change
  14    0.000       Param Server   JSH AT       MAC Request Initialization
  15    0.000       Param Server   JSH AT       MAC Request Initialization
  16    0.000       Param Server   JSH AT       MAC Request Initialization
  17    0.000       Param Server   JSH AT       MAC Request Initialization
  18    0.005       NetWare        JSH AT       NCP C Find nearest file server
  19    0.002       JSH AT         NetWare      XNS Uninterpreted packet type: 0 (Novell Netware (Unknown))
  20    0.004       NetWare        JSH AT       XNS RIP request
  21    0.000       Broadcast      NetWare      MAC Standby Monitor Present
  22    0.002       JSH AT         NetWare      XNS Uninterpreted packet type: 0 (Novell Netware (Unknown))
  23    0.003       NetWare        JSH AT       NCP   Create Connection
  24    0.002       JSH AT         NetWare      NCP R OK
  25    0.003       NetWare        JSH AT       NCP C Propose buffer size of 1024
  26    0.001       JSH AT         NetWare      NCP R OK Accept buffer size of 1024
  27    0.003       NetWare        JSH AT       NCP C Logout
  28    0.007       JSH AT         NetWare      NCP R OK
  29    0.003       NetWare        JSH AT       NCP C Get server's clock
  30    0.002       JSH AT         NetWare      NCP R OK
  31    0.365       NetWare        JSH AT       NCP C Set login dir to MSDOS
  32    0.006       JSH AT         NetWare      NCP R No such object
  33    0.011       NetWare        JSH AT       NCP C Get station number
  34    0.001       JSH AT         NetWare      NCP R OK Station is 01
  35    0.471       NetWare        JSH AT       NCP C End of task
  36    0.002       JSH AT         NetWare      NCP R OK
  37    0.047       NetWare        JSH AT       NCP C End of task
  38    0.002       JSH AT         NetWare      NCP R OK
  39    0.101       NetWare        JSH AT       NCP C End of task
  40    0.002       JSH AT         NetWare      NCP R OK
  41    0.074       NetWare        JSH AT       NCP C End of task
  42    0.002       JSH AT         NetWare      NCP R OK
  43    0.075       NetWare        JSH AT       NCP C Get dir path of handle 01
  44    0.002       JSH AT         NetWare      NCP R OK Path=SYS:LOGIN
  45    0.004       NetWare        JSH AT       NCP C Dir search parms for [null]
  46    0.002       JSH AT         NetWare      NCP R OK Next=-1
  47    0.003       NetWare        JSH AT       NCP C Dir search LOGIN.???
  48    0.002       JSH AT         NetWare      NCP R OK File=LOGIN.COM
  49    0.003       NetWare        JSH AT       NCP C Get dir path of handle 01
```

Figure 7-6: Sniffer Protocol Summary Printout

The control field contains detailed information on how to handle the LLC frame itself. There are Information (I), Supervisory (S), and Unnumbered (U) LLC frames. NetWare uses the simplest of them, the U frame that is used for Type 1 operation (see Chapter 3). The I frame is used for connection-oriented service, Type 2, in which sequence numbers are used at the LLC level. NetWare uses sequence numbers at the transport level to ensure that packets are not lost or replicated.

If the DSAP is E0, then The Sniffer interprets the frame as a NetWare frame and will treat the rest of the frame accordingly. When examining a captured frame, three views are available to the user -- a summary view, a detailed view, and the hex view. When inside one of the view windows, the cursor can be moved around and the other windows will change accordingly. For example, one can select the summary window to display only the highest protocol recognized by The Sniffer within the frame or select the Summary encapsulated view. Then, by moving the cursor to, say the NetWare NCP portion of the frame in the summary window, the Detail window now contains an English explanation of the NCP request and the Hex window highlights the actual location of the bits in the frame along with their ASCII (or EBCDIC) representation.

All NetWare Core Protocols are carried across the Token-Ring using the same basic format. This format, created from actual Sniffer observation along with accompanying IBM, IEEE, Xerox, and Novell documentation, is illustrated in Figure 7-7. The standard XNS Internet packet format is used by Novell but not the standard XNS sequenced packet protocol found in the Internet data field -- replaced instead by NetWare Core Protocol (NCP) that contains its own Novell-defined fields, including a sequence number field.

We now realize that A) NetWare is not purely XNS and B) the dialogue between a workstation and a file server covers only four layers of protocol, not the full seven of the OSI reference model. The four layers are: physical, data link, network, and transport. The transport level really contains elements of a session and application layer since this is the last layer that the file server interprets.

For serious protocol development and monitoring or determining problems on large LANs, a protocol analyzer such as The Sniffer may be for you -- but be prepared to invest at least $20,000. If only error monitoring and packet counting is desired, then the IBM Token-Ring Network Manager may be a more worthwhile investment. But then you may want your Token-Ring to reveal all!

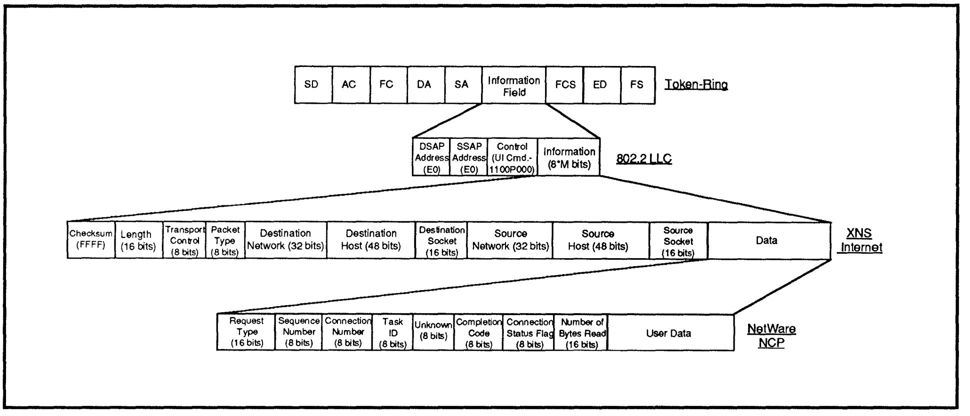

Figure 7-7: Encapsulated NetWare Protocols

Chapter 8

Performance

Trying to gauge performance of any network is difficult due to the number of variables involved. Rather than attempt an exhaustive performance analysis of the Token-Ring (which would call for another report written about the subject), a number of simple case studies will be examined to at least get a "feel" for the overall throughput capability.

Loading

IBM has completed a number of studies to determine Token-Ring performance with the PC Adapter. For comparison purposes, tests were also performed on the PC Network. The reader is forewarned that the numbers given are preliminary and not guaranteed by IBM.

Assuming the network could be loaded with frames as close to 100% as possible, the Token-Ring can carry (at the physical link level) data at 3.7 Mbps, or about 92.5% of the 4-Mbps bandwidth. maximum data capacity of PC Network is 1.5 Mbps, or about 75% of the 2-Mbps bandwidth. This means that if the device attached to the Token-Ring (such as a PC) can sustain back-to-back transmission of packets, the Token-Ring will be more than a factor of two better in performance than PC Network. This is due to the increased overhead, such as preambles and large interframe spacing time present in the carrier sense multiple access (CSMA) protocol of PC Network.

NETBIOS

The throughput was then measured over multiple sessions at the NETBIOS level, using 16 Kbyte messages. With PC Network, the maximum possible throughput at the session level was found to be 650 Kbps. Using the Token-Ring with the NETBIOS emulator running in an AT, the throughput was found to be 2.0 Mbps, a factor of three better than PC Network. Note that the PC Adapter card for the Token-Ring uses shared memory. Another factor which slows down NETBIOS is that every packet is acknowledged, whereas on the Token-Ring, a parameter can be set to acknowledge every N packets.

IBM discovered that the maximum data rate of any one station on the Token-Rings depends on how fast the "engine" (processor) of that station is. For example, a single AT can drive the network at nearly double the rate of a standard PC or XT. Figure 8-1 shows the throughput at the data link (LLC) level for a PC XT, with various packet sizes (2 Kbytes is the maximum link packet size of the 4-Mbps Token-Ring). The single link refers to one PC sending to one other PC. Multiple links refer to multiple PCs sending to one other PC. The test was repeated for the PC AT; its results are shown in Figure 8-2.

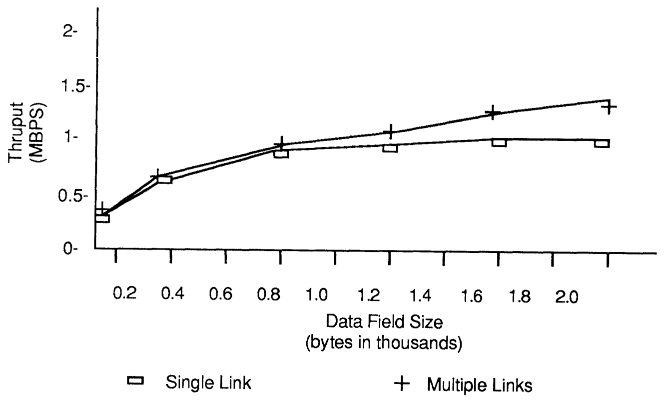

Figure 8-1: Ring Throughput for PC XT

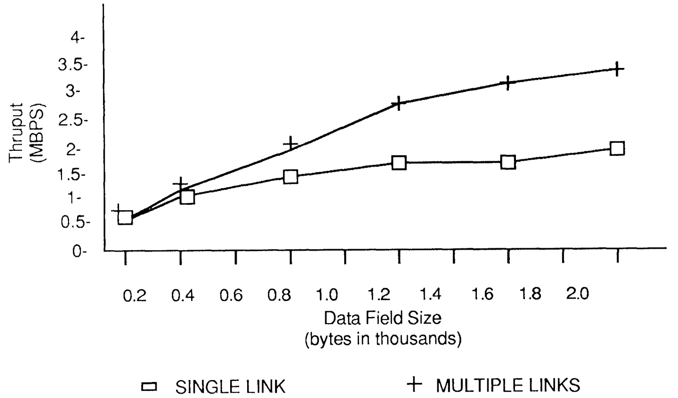

Figure 8-2: Ring Throughput for PC AT

"Real-world" tests were also conducted using the Token-Ring with the NETBIOS emulator and the PC LAN Program 1.0. One PC AT was configured as a dedicated file server. A 40 Kbyte file was loaded by one to four workstations. The response time experienced by the workstations is illustrated in Figure 8-3. The top shaded portion of the bar is the time spent in the AT servicing (mainly fetching data from the hard disk) the request. The bottom portion is the actual time spent transferring the data over the ring. As with all PC local area networks, the response time will depend not only on the performance of the network, but also on how fast the server can service the request. For file servers, a limiting factor is how fast the hard disk is. The trick for optimum performance is to utilize as much bandwidth as possible.

Host Processor

Another interesting way to look at performance is by the percent utilization of the PC's processor while performing a network operation. This indicates how much the processor "idles" (or the time wasted) in which a multi-tasking operation could be implemented. The following are results of the IBM study:

40 Kbyte File load from Server

PC Type	LLC	NETBIOS
PC or XT	<8%	<20%
AT	<3%	<8%

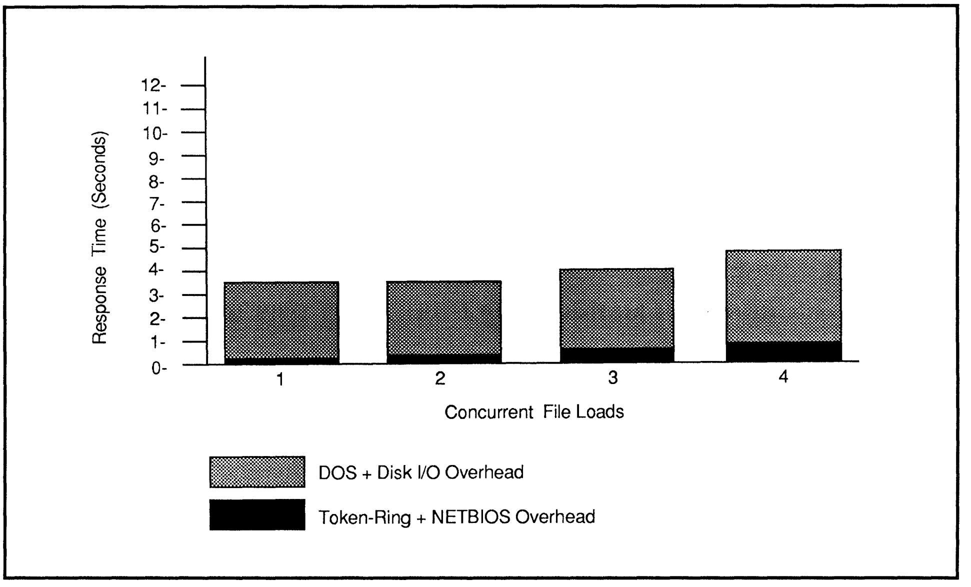

Figure 8-3: File Load Times

100 Kbyte File Copy from/to Server

PC Type	LLC	NETBIOS
PC or XT	<21%	<59%
AT	<8%	<21%

The general rule of thumb to maximize performance of the ring, regardless of the PC type, is to minimize links and maximize the buffers.

Generating Traffic

Using the traffic generation feature of a LAN monitoring tool such the Sniffer, one can test a token-ring under a constant load. The Sniffer allows you to specify the number of bytes to send, who to send it to (or all stations broadcast), and how much delay (in milliseconds) there is between sending frames. When the frames are being sent, frame counts, and Kbytes per second transmission throughput are displayed in real-time.

When specifying the maximum token-ring packet size (4048 bytes) with no delay between sending, the Sniffer was able to sustain a transmission rate of approximately 330 Kbytes per second, or 2.64-Mbps which consumes about 66 percent of the 4-Mbps token-ring bandwidth. When performing various DOS commands from a single

AT to NetWare, very little degradation was noticed. This is due to several factors: fair token access, only three stations on this particular test network, and the overhead at the workstation and server required to actually carry out the request.

To get a feel for the token passing overhead, the packet size was reduced to 18 bytes. With no delay between sending, The Sniffer was able to send approximately 20 Kbytes per second. Once again, this shows that the more data sent in a frame, the more efficient utilization of the network. Of course, for fairness to others trying to access shared resources like a file server, the vendor of that server will usually put a limit on the number of data bytes transferred at a time in order to service requests in a round robin fashion. For example, The Sniffer showed that with NetWare, 1024 bytes are transferred during the loading or copying of DOS files.

One could also use the transmit feature to see how a station behaves when receiving packets "out of the blue." In addition to specifying the destination address, one can also set the first 16 bytes of the frame to meet a certain protocol criteria.

Novell Benchmarks

Anytime a LAN vendor (or an "independent" magazine for that matter) publishes benchmarks for various LANs, it is subject to controversy. The following comments and graphs are based on the Novell LAN Evaluation Report published back in 1986, but still valid today for comparison purposes -- the benchmarks chosen for inclusion here were selected on the basis of comparing performance of various LAN hardware. NetWare operates with several different LAN adapters. This should give the reader a rough rule of thumb for judging relative LAN performance based on the underlying transport mechanism; the benchmarks are in no way intended to give exact performance promises in real-world implementations.

One such benchmark was designed to determine the maximum bandwidth, in kbytes per second, that each network could support. 4096 bytes of data was read 100 times from a NetWare server -- thus the data was always in the server's cache memory. Four IBM PC AT workstations (on the token-ring) and six IBM PC AT workstations (for all other LANs) were used to drive the network to its maximum traffic capability. Apparently Novell did not have six token-ring adapters on hand and felt that the ATs could sufficiently load the network even with four -- we can only speculate that the actual token-ring throughput may have been a bit higher if six adapters were used.

As we can see from Figure 8-4, all LANs but the one using the smart Ethernet adapter from 3com were slower than the Token-Ring. Even though Ethernet has more than twice the transmission speed (10 vs. 4-Mbps), token-ring was not too far behind in available working bandwidth.

Conclusions from all this are hard to draw. There are so many factors that make it difficult to compare different LAN technologies.

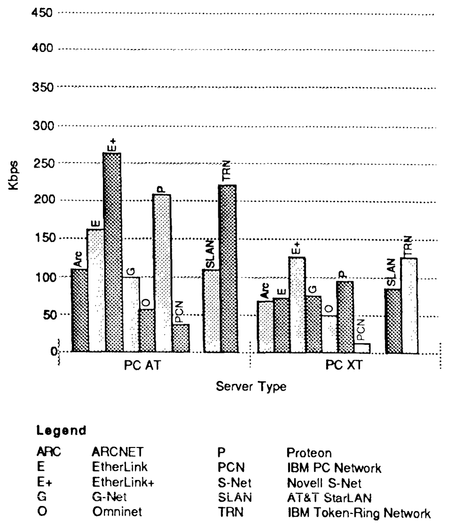

Figure 8-4: LAN Working Bandwidth

Some of these factors include how well the interface between the host and the adapter works (i.e., how fast can frames be moved to/from the adapter to host memory) and intelligent adapters with on-board processors vs. "dumb" adapters with little or no frame processing capability (witness the fact that the 3Com Etherlink+ adapter with its on-board processor performs much better than the standard Etherlink adapter).

16-Mbps

Benchmarking 16-Mbps token-ring has been difficult. Many recent benchmarks in recent popular magazines have shown little improvement over 4-Mbps token-rings or Ethernet. Much of this is attributable to the fact that these benchmarks include not only the adapter, but also the adapter handlers (drivers), protocol stacks, NETBIOS interfaces, PC-DOS, etc. Another problem is that in many cases, the adapter handlers are "universal" for portability reasons, and don't take advantage of enhanced features on newer adapters.

What we really need is testing as close to the adapter as possible, eliminating the higher level protocols, operating systems, and applications. This way, we can determine where the real bottlenecks are and improved our drivers, applications, etc.

To get a feel for "Raw" token-ring performance, TI has done performance modeling of a token-ring using the TMS380 chip set. The model was verified against an actual 4-Mbps token-ring adapter using the TI chip set, running in an IBM PC AT. Since the model was deemed accurate given the parameters of the AT and the TMS380, the token-ring data rate parameter was increased to 16-Mbps. The resultant transmit throughput is shown in Figure 8-5. With this higher-speed token-ring, the engine begins to become a limiting factor. It took a 10 Mhz AT with a 1K buffer and a frame size of 4 Kbytes, to achieve the indicated throughput of 12 Mbps. Thus the newer generation of PCs -- the Personal System/2 with 80286 and 80386 processors -- should be able to drive the next generation of token-rings (16 Mbps) to maximum throughput levels. This is especially true in lieu of the fact that the PS/2 model 50 and higher incorporates the Micro Channel which has much faster direct memory access (DMA) rates than the AT. This will help designs using the TMS380 chip set since it relies on DMA (as opposed to the faster host/adapter memory-mapped buffers as used in IBM's Token-Ring adapter design) for its operation.

Trace and Performance Program

IBM is offering a program that will measure actual performance of a Token-Ring. The IBM Token-Ring Network Trace and Performance Program is a menu- driven program product that, by working in conjunction with the IBM Token-Ring Network Trace and Performance Adapter II or the IBM Token-Ring Network Trace and Performance Adapter/A, analyzes traffic activity handled by a local area network and also performs user data throughput measurements on IBM Token-Ring Networks. The trace function analyzes application programs using different protocols on a Token-Ring; the performance function monitors the use of the Token-Ring by all or a subset of stations.

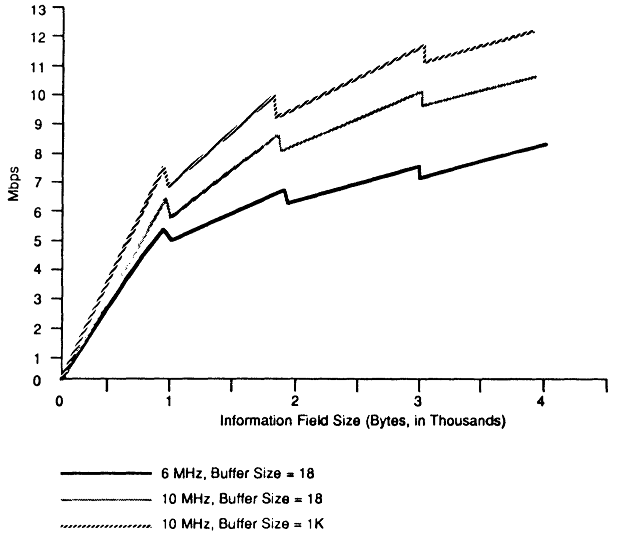

Figure 8-5: Ring Throughput at 16 Mbps

The program along with the respective adapter offer the following capabilities:

- Provide a window to the traffic on the Token-Ring using the trace facility
- Enable analysis of Token-Ring traffic using the trace facility
- Help in understanding of displayed/printed frames by interpreting the supported protocol fields into readable English
- Enable trace activity to be triggered by user-specified data pattern or time of day
- Display realtime Token-Ring use percentage and average waiting time
- Analyze use of data based on time interval, frame size, and byte count
- Display traffic statistics in graph and table formats

PLAN/36 Performance Tool

The IBM Personal Computer PLAN/36 Performance Tool is a PC program that supports IBM System/36 (5360, 5362 and 5364), IBM System/38 and the IBM Token-Ring Network (a LAN connected to the System/36). It can be used to evaluate new applications, existing workloads or workload growth by showing whether the system meets the performance requirements. PLAN/36 can also be used to project long-range systems requirements based on a specified growth rate.

The input to PLAN/36 for System/36 and System/38 includes:

- Workload descriptions for interactive applications
- Workload descriptions for batch applications and spool
- Measured workload data
- Performance Requirement Objectives (PRO) for these workloads

Output from PLAN/36 for System/36 and System/38 includes:

- Expected throughput
- Average response time
- Resource utilization estimates
- Recommended hardware configuration

The input to PLAN/36 for the Token-Ring Network LAN on System/36 includes:

- LAN-attached system response file names to be modeled
- Cabling type
- Background network workloads

Output from PLAN/36 for the Token-Ring Network LAN on System/36 includes:

- Token-Ring Network utilization-per-transaction for each application
- Communication delay-per-transaction for each application

PLAN/36 can be used as a:

- Performance Tuning Tool: Using System Measurement Facility (SMF) as input, PLAN/36 helps to identify resource constraints and evaluates alternative solutions.

- Performance Prediction Tool: As a prediction tool PLAN/36 will evaluate the impact of adding a new application or increasing the number of users of the current application mix.
- System Management Tool: Given an approximate growth rate for the current workload, PLAN/36 will identify which resource may need to be upgraded and when.
- System/36 to System/38 Migration Workload Migration Tool: PLAN/36 has the capability of approximating the System/38 model needed to accommodate most existing System/36 workloads.

PLAN/36 is a combination of an analytic model and an expert system. As such, the results depend on the accuracy of the input, the analytic model and the rule base for the expert.

Appendix A: Acronyms

AC	access control
ACDI	Asynchronous Communications Device Interface
ACF	Advanced Communication Facility
ALS	Advanced Low-Power Schottky
ANSI	American National Standards Institute
API	application programming interface
APPC	Advanced Program-to-Program Communication
APPC/PC	APPC for the Personal Computer
ARCNET	Attached Resource Computer Network
ARP	Advanced Research Project
ASCII	American Standard Code for Information Interchange
ASIC	application specific integrated circuit
ASYNC	asynchronous
AWG	American Wire Gauge
BIOS	basic input output system
BISYNC	bisynchronous
BIU	bus-interface unit
BNC	Banana Nut Connector
C&SM	Communications and Systems Management
CBX	computerized branch exchange
CCB	command control block
CCITT	International Telegraph and Telephone Consultative Committee
CD	collision dectection

CDMS	Cable Data Management System
CICS	customer information control system
CMOS	complementary metal oxide semiconductor
CMSA	Carrier Sense Multiple Access
CNM	Communication Network Management
CP	Communications Processor
CPU	central processing unit
CRC	cyclic redundancy check
CSMA/CD	carrier sense multiple access/collision detection
DB/2	Database/2
DCA	document content architecture
DDN	defense data network
DFT	distributed function terminal
DIA	data interchange architecture
DIF	data interchange format
DISOSS	Distributed Office Support Systems
DIX	DEC Intel Xerox
DLC	data link control
DMA	direct memory access
DOS	disk operating system
DSAP	destination service access point
DWC	Distributed Wiring Concentrator
EBCDIC	extended binary-coded decimal interchange
ECF	Enhanced Connectivity Facility
ED	ending delimiter
EHLLAPI	Emulator High level Language Application Program Interface
EISA	Enhanced Industry Standard Architecture
EMI	electro-magnetic interference
EPROM	erasable programmable read-only memory

FAT	file allocation table
FCC	Federal Communications Commission
FDDI	Fiber Distributed Data Interface
FCS	frame check sequence
FEP	Front End Processor
FSD	File System Driver
GDDM	Graphical Display Data Manager
GDS	General Data Stream
HDLC	high level data link control
ID	IDentification
IDNX	Integrated Digital Network Exchange
IEEE	Institute of Electrical and Electronics Engineers
I/O	Input/Output
IFIP	International Federation for Information Processing
IPC	interprocess communication
ISA	Industry Standard Architecture
ISO	International Organization for Standardization
Kbps	thousand bits per second
Kbytes	thousand bytes
LAB	Line Attachment Base
LAN	Local Area Network
LCC	logical link control
LED	light emitting diode
LEN	Low Entry Networking
LLC	logical link control
LPDU	logical link protocol data unit
LSAP	link service access point

LU	Logical Unit
MAC	media access control
MAP	Manufacturing Automation Protocol
MAU	multistation access unit
Mbps	million bits per second
Mbytes	million bytes
MCB	message control block
MDR	Media Driver/Reciever
MEU	memory-expansion unit
MIC	media interface connector
MMIO	memory mapped i/o
MSB	most significant bit
MS-DOS	microsoft disk operating system
MVID	major vector ID
MVS	Multiple Virtual System
NAUN	nearest active upstream neighbor
NCB	network control block
NCP	network control processor
NEBEUI	NETBIOS User Interface
NETBIOS	Network Basic Input Output System
NIU	network interface unit
NMC	Network Management Console
NMPM	Network Management Performance Monitor
OCCF	Operator Communications Control Facility
OS/2	Operating System/2
OS/2 EE	Operating System/2 Extended Edition
OSI	Open Systems Interconnection
PBX	Private Branch Exchange

PC	Personal Computer
PC-DOS	Personal Computer Disk Operating System
PDS	Personal Decision Series
PDU	Protocol Data Unit
PHY	PHYsical
PIO	programable I/O
POST	power-on-self-test
PPP	priority bits (3)
PS/2	Personal System/2
PU	Physical Unit
QFP	quad flat packs
QMF	Query Management Facility
RAM	random access memory
RAS	reliability, availability, serviceability
REM	Ring Error Monitor
RES	Configuration Report Server
ROM	read-only memory
RPL	remote program load
RRR	reservation bits (3)
RT PC	risc technology personal computer
SAA	systems application architecture
SABME	set asynchronous balanced mode extended
SAP	service access point
SD	starting delimiter
SDLC	synchronous data link control
SFAP	structured filed and attribute processing
SIF	System Interface
SMB	server message block
SMF	System Measurement Facility

SMT	station management
SNA	systems network architecture
SNADS	SNA distribution system
SQL	sequential query language
SPCF	Service Point Command Facility
SPCS	Service Point Control Service
SRAM	Static Random Access Memory
SRPI	server-requester protocol interface
SSAP	Source Service Access Point
SSCP	System Services Control Point
STP	shielded twisted-pair
TCP/IP	Transport Control Protocol/Internet Protocol
TIC	token interface coupler
TSO	time share option
UTP	unshielded twisted-pair
VGA	Video Graphics Array
VLSI	very large scale integration
VM	virtual machine
VTAM	virtual telecommunications access method
XID	eXchange IDentification
XNS	Xerox Network Systems

Appendix B: References

Architecture Technology Corporation publications may be obtained by calling (612) 935-2035.

IBM publications should be ordered through your local representative.
IEEE publications can be ordered from The Institute of Electrical and Electronics Engineers, Inc., 345 East 47th Street, New York, NY 10017.

Texas Instruments publications may be obtained from by calling (800) 232-3200.

1. Architecture Technology Corporation, The Ring-Based Local Networks Report
2. Architecture Technology Corporation, Token-Ring Technology Report.
3. Haugdahl, J. Scott, Analyzing Network Traffic, PC Tech Journal, Volume 5 Number 10, October 1987, pp. 48-62.
4. Haugdahl, J. Scott, Inside NETBIOS, Architecture Technology Corporation.
5. Haugdahl, J. Scott, Inside SAA, Architecture Technology Corporation.
6. Hurwicz, Michael, Inside APPC, Architecture Technology Corporation.
7. IBM Corporation, An Introduction to Advanced Program-to-Program Communication (APPC), GG24-1584.
8. IBM Corporation, Token-Ring Network Architecture Reference, SC30-3374-01.
9. IBM Corporation, Token-Ring Network Introduction and Planning Guide, GA27-3677-0.
10. IBM Corporation, IBM Token-Ring Network Administrator's Guide, GA27-3748.
11. IBM Corporation, Token-Ring Network PC Adapter Technical Reference, SC30-3383.
12. IBM Corporation, IBM Token-Ring Network Bridges and Management, GA24-3062.
13. IBM Corporation, IBM Token-Ring Network Telephone Twisted-Pair Media Guide GA27-3714-0.
14. IBM Corporation, IBM Token-Ring Network--Guide to Small Networks 61X3812.
15. IBM Corporation, IBM Token-Ring NetworkInstallation Guide GA27-3678.
16. IBM Corporation, Local Area Network Technical Reference, SC30-3383-2.
17. IBM Corporation, IBM NETBIOS Application Development Guide 68X2270.
18. IBM Corporation, Systems Network Architecture Introduction, GA27-3116.
19. IBM Corporation, SNA Reference Summary, GA27-3136.
20. IBM Corporation, SNA Technical Overview, GC30-3073.
21. IBM Corporation, SNA Format and Protocol Reference Manual: Architecture Logic for LU Type 6.2, SC30-3269.
22. IBM Corporation, Systems Application Architecture Overview, GC26-4341-0.

23. IEEE, An American National Standard--IEEE Standards for Local Area Networks: Token-Ring Access Method and Physical Layer Specifications (ANSI/IEEE STD 802.5-1985 or ISO Draft Proposal 8802/5)
24. IEEE, An American National Standard--IEEE Standards for Local Area Networks: Logical Link Control (ANSI/IEEE Std 802.2-1985, or ISO Draft International Standard (DIS) 8802/2)
25. IEEE 802.5 Committee, 802.5B Recommended Practice for Use of Unshielded Twisted Pair Cable (UTP) for Token-ring Data at 4 Mbits/sec
26. IEEE 802.5 Committee, 802.5C Dual Ring Operation with Wrapback Configuration
27. IEEE 802.5 Committee, 802.5D Enhancement for Multiple Ring Networks (IBM source routing)
28. IEEE 802.5 Committee, 802.5E Management Entity Specification
29. IEEE 802.5 Committee, 802.5F Enhancement for 16 Mbits/sec Operation
30. IEEE 802.5 Committee, 802.5H Recommended Practice for LLC Type 3 Support
31. IEEE 802.5 Committee, 802.5I Enhancement for Early Token Release
32. IEEE 802.5 Committee, 802.5L Token-ring Standard Maintenance (corrections and clarifications in the 1985 802.5 standard)
33. Novell, Inc., LAN Operating System Report 1986, October 1986.
34. Novell, Inc., LAN Evaluation Report 1986, October 1986.
35. Texas Instruments, Product Description: The TMS380 LAN Adapter Chip Set (SPWV001)
36. Texas Instruments, TMS380 Adapter Chipset User's Guide (SPWU001D)
37. Texas Instruments, TMS380 Adapter Chipset User's Guide Supplement (SPWU003)

Appendix C: Token-Ring Vendors and Products

Adapters

4-Mbps, 8-bit PC/XT Bus	Andrew Corporation (Torrance, CA) Cisco (Menlo Park, CA) DSC Communications (Santa Clara, CA) Gateway Communications (Irvine, CA) Hughes LAN Systems (Mountain View, CA) IBM (White Plains, NY) Madge Networks (San Jose, CA) NCR (Dayton, OH) Olicom (Denmark) Proteon (Westboro, MA) Racore (Los Gatos, CA) 3Com (Santa Clara, CA) Tiara (Mountain View, CA) Ungermann-Bass (Santa Clara, CA) Unisys (Blue Bell, PA) Western Digital (Irvine, CA)
4-Mbps, 16-bit PC/AT Bus	Cisco Data General (Westboro, MA) Gateway Communications Hughes LAN Systems IBM Lantana Madge Networks Olicom Proteon PureData (Richmond, Ontario) Racore 3Com Tiara
4-Mbps Micro Channel	Cisco IBM

	Madge Networks Proteon PureData Racore
4/16-Mbps, 8-bit PC Bus	IBM
4/16-Mbps, 16-bit PC/AT Bus	IBM Lantana Madge Networks NCR Olicom Proteon
4/16-Mbps Micro Channel	IBM Lantana Madge Networks
4-Mbps Eclispe	Data General
4/16-Mbps EISA	Madge Networks Proteon
4-Mbps Macintosh	Apple (Cupertino, CA) Asante Technologies (Sunnyvale, CA)
4-Mbps Multibus	Netronix (Petaluma, CA)
4-Mbps Q-bus	Simpact (San Diego, CA)
4-Mbps VMEbus	Formation (Mt. Laurel, NJ) Interphase (Dallas, TX)
4-Mbps IBM FEP	IBM
4/16-Mbps IBM FEP	IBM
4-Mbps AS/400	IBM
4-Mbps Series/1	IBM
4/16-Mbps RISC/6000	IBM
Bridges	
Token-Ring-to-Token-Ring	CrossComm (Marlboro, MA) IBM Madge Networks Ungermann-Bass

Token-Ring-to-Ethernet	Chipcom CrossComm IBM Ungermann-Bass
Token-Ring-to-FDDI	Chipcom
Token-Ring-to-Localtalk	Cayman Systems (Cambridge, MA)
Token-Ring-to-Broadband	Netronix
Token-Ring-to-Remote-Token-Ring	Cisco CrossComm Eicon Halley Systems (San Jose, CA) Microcom (Norwood, MA) T3 Technologies (Research Triangle Park, NC) Vitalink
Chip Sets	
4-Mbps	TI (Dallas, TX)
4/16-Mbps	TI Toshiba (Irvine, CA)
Gateways	
3270	Adacom Apple AST Research Attachmate Banyan DCA IBM ICOT Network Systems (Mobile, AL) Microcom NCR Compten (St. Paul, MN) NSA (Lajuna Hills, CA) Novell Rabbit Software Relay Communications Software Dynamics (Dunedin, FL) 3Com Trisystems

	Ungermann-Bass Waterloo Microsystems
3174	Attachmate Harris IBM Memorex Telex Nokia Data
Media Tester	Tektronix (Redmond, OR)
Multi-station Access Units (MAUS)	Andrew Corporation Belkin Components (Gardena, CA) Data General DSC Communications Gateway Communications General Technology (Melbourne, FL) Hughes LAN Systems IBM Lynn products (Torrance, CA) Madge Networks Mitsubishi Cable America (New York, NY) Nevada Western (Sunnyvale, CA) NCR Olicom Onan (Minneapolis, MN) Optical Data Systems (Richardson, TX) Proteon Racore RadCom (Rochelle Park, NJ) 3Com STAR-TEK (Northboro, MA) Thomas-Conrad (Austin, TX) Ungermann-Bass
MAUS Supporting Fiber Optics	Ericsson (Sweden) Optical Data Systems Siecor Corporation (Research Triangle Park, NC) Versitron (Annapolis Junction, MD)
MAUs Supporting 16 Mbps Over Unshielded Twisted-Pair	Cabletron (Rochester, NH) MUX-LAB (Menlo Park, CA) Proteon STAR-TEK Ungermann-Bass

Protocol Analyzers	Novell/Excelan Network General Vance Systems (Chantilly, VA)
Repeaters	
4-Mbps Copper	Andrew Corporation Data General IBM Madge Networks STAR-TEK
4-Mbps Fiber	Data General IBM RadCom Raycom Systems (Boulder, CO) Siecor Corporation S.I. Tech (Geneva, IL)
16-Mbps Fiber	IBM RadCom S.I. Tech
Laser	Laser Communications (Lancaster, PA)
Microwave	Motorola (Schaumburg, IL)
Routers	Cisco CrossComm Eicon Technology Halley Systems IBM
File Servers	Artisoft Banyan CBIS Corvus DCA IBM NCR Novell Performance Technology 3Com Torus Systems Ungermann-Bass Waterloo Western Digital

Glossary

ACF/NCP	Resides in the IBM 3720 or IBM 3725 Communication Controller and provides the physical management of the communication network. Its main function is to control attached lines and workstations, perform error recovery, and route data through the SNA network.
ACF/VTAM	The base for the IBM SNA network, which may be thought of as an "operating system" for the network. Its functions are analogous to the functions of an IBM host operating system in terms of resource sharing and logical handling of user requests.
Advanced Program-to-Program Communication (APPC)	An architecture for peer-to-peer, application-to-application program. Also called LU6.2, which is the technical name for the marketing name APPC.
alert	The main network management message for forwarding problem determination information to a network operator.
Application Programming Interface (API)	A protocol boundary which can be used by arbitrary user-written programs.
Advanced Peer-to-Peer Networking (APPN)	An extension of LU6.2 and PU2.1, which allows peripheral nodes to perform intermediate and dynamic routing functions.
architecture	A formal set of definitions describing how components of an overall system must interact to work for a common objective. Architecture is of a general nature, not tied into a specific product implementation.
common communications support	Establish SNA and international communications standards chosen for SAA to provide enterprise-wide systems solutions.
common programming interface	Building blocks for architecture development under SAA that provide a consistent programmer view for interfaces, languages, and program services.
conversation	A logical connection between two transaction programs.

cooperative processing	A loosely defined term that includes extensions of resources (virtual disks and printers), access to distributed data, and offloading of processing in distributed applications.
datastream	A series of bits appearing on a communications link structured according to some agreed rules. These rules may consist of character codes, control characters, header and trailer information, and field lengths.
Document Content Architecture (DCA)	Defines the form and meaning of the content of a document. Both revisable form and final (printable) forms are defined.
Document Interchange Architecture (DIA)	Defines how document distribution and processing functions are to be communicated through an office system network. There are three categories of service: library services, distribution services, and application processing.
Enhanced Connectivity Facilities (ECF)	A uniform architecture for PCs/Personal Systems to exchange data with Sytems/370 host processors and to provide access to resources on these processors. Users can access host files, disk space, and printer facilities.
Enterprise	A business organization.
Logical Unit (LU)	Code residing in a machine on a network, either in firmware or in software, which allows the machine to communicate on the network. The LU provides services to the end-user, and may be thought of as the end user's "port" into the network. Nodes may contain multiple LUs.
Low Entry Networking (LEN)	Direct peer-to-peer connections between PU2.1 nodes and attachment to System/36 APPN. LEN is the marketing name for PU2.1 which is the technical name.
NETBIOS	Network Basic Input Output System. A collection of interfaces and protocols designed to support many of the functions of session management as defined by the OSI reference model. Also includes datagram support.
node	A machine at the end of a link, or at the intersection of two or more links on an SNA network. Terminals, communications controllers, and host processors are all nodes.
Operating System/2 (OS/2)	Designed for IBM's Personal System/2, OS/2 is a multi-tasking operating system that operates in the protected mode of 80286 or

80386 processors. OS/2 enhances the functionality of DOS and offers an extended version to support SAA.

Open Systems Interconnect (OSI) A seven-layer international reference model from which to develop protocols and standards to support non-homogeneous distributed systems operating across a wide variety of networks.

peer Generally, being equal. Used for distributed systems and intelligent work stations to denote a balanced relationship in their communication (no master/slave or primary/secondary relationship).

peer-to-peer communication Communication between two LUs without an intermediate host.

Physical Unit (PU) A component of an SNA node that manages network resources, such as communication lines. There is one PU per node.

protocol An agreed-upon method between two parties for controlling an orderly flow of information between them. This includes a common understanding of related syntax and semantics.

Server-Requester Programming Interface (SRPI) A programming interface for developing applications that require coordination between System/370 hosts (servers) and personal computers (requesters) allowing users to query and extract data from hosts, transfer files between hosts and PCs, and issue host commands from a PC.

session A logical connection between two NAUs.

Structured Query Language (SQL) A high-level non-navigational language that allows programmers to access a relational database such as DB2.

Synchronous Data Link Control (SDLC) A low-level communications protocol which can support APPC.

Systems Application Architecture (SAA) A collection of selected software interfaces, conventions, and protocols that is the framework for development of consistent applications across the future offerings of the major IBM computing environments -- System/370, System/3X, and Personal Computer.

Systems Network Architecture (SNA) IBM's master plan for allowing its products to communicate. SNA defines logical structures, formats, protocols, and procedures for exchanging information on a data communications networks.

SNA Distribution System (SNADS) An architecture incorporating DCA, DIA, and APPC that supports store-and-forward document distribution between multiple mainframes running DISOSS.

token-ring IBM's strategic local area network (LAN) based on a star-shaped ring wiring topology using a controlled token for access. Also an IEEE 802.5 standard.

transaction A logical unit of work, from the point of view of an application program on a network.

transaction program An application program which performs transactions in cooperation with one or more other application programs on a network.

user A person or application program requiring the services of a computing system.

user support system Provides languages and an architected set of services and interfaces for SAA which allow integrated application development, as well as flexible placement of function and data.

Virtual Machine (VM) The functional simulation of a computer and its associated input/output devices.

X.25 A CCITT standard consisting of a collection of physical, datalink, and network protocols for public long-haul networks.

Index

Numeric

A

N

O

P

U

V

W

X

About the Author...

J. Scott Haugdahl is a Senior Communications Engineer at Architecture Technology Corporation. His work has included simulation and performance analysis of multiprocessor computer systems, design and implementation of specialized server software and protocols for personal computer local networks, design and testing of advanced hardware for servers of PC local networks, and analysis of the latest developments and product offerings in the local-area network marketplace. He has researched, written, and presented numerous seminars on LANs in both the U.S. and Europe. He has also presented papers for IEEE and CW Communications conferences and is the author of the books *Inside the Token-Ring, Inside NETBIOS,* and *Inside SAA*. Mr. Haugdahl received his B.S. in Computer Science from the University of Minnesota, Institute of Technology.

Other Books of Interest from

Architecture Technology Corporation

Inside SAA

SAA (Systems Application Architecture), is IBM's set of architectures for implementing distributed computing and data systems across its strategic family of computers; the Personal System/2, Systems 3X, and the System/370. Written by J. Scott Haugdahl, this book begins with a historical and conceptual overview of SAA, including its relationship to IBM's SNA (Systems Network Architecture) and the benefits it offers to vendors and users. Subsequent chapters discuss the key elements of SAA in more detail -- including Low Entry Networking (LEN), Advanced Program to-Program Communication (APPC), Distributed Data Management, Enhanced Connectivity Facilities (ECF), Integrated Office Systems, Network Management with discussion of NetView, and the SAA elements in Operating System/2 Extended edition.

Inside APPC

APPC (Advanced Program-to-Program Communications) is an architecture destined for implementation on an extremely wide range of IBM and non-IBM datacom products. APPC is the architecture most central to IBM's distributed processing strategy. Written by Michael Hurwicz, this book begins with a conceptual and historical overview of APPC, continues with coverage on: LU6.2 basics -- states, functions, services, and relationship to PU2.1; the LU6.2 architecture and formats; SNA Service Transaction Programs (TPs) -- SNADS, DIA, and DDM; and concludes with detailed examples of how APPC verbs are used.

Inside NETBIOS

NETBIOS, a high-level programming interface to IBM local networks, received a new lease on life when IBM added a NETBIOS emulator to its Token-Ring. Written by J. Scott Haugdahl, *Inside NETBIOS* provides an in-depth look at the protocols that are shaping PC LAN applications development. Coverage begins with introductory material covering NETBIOS history and protocol layers, continuing with programming procedures which includes a detailed description of NETBIOS commands and a discussion of the implementation differences between PC Network and Token-Ring. The Server Message Block and file access, printer server, and message protocols are covered in detail. In addition, selected non-IBM implementations of NETBIOS are examined.

ISBN 0-939405-00-8