Systems Network Architecture

(SNA) Networks

Edwin R. Coover

IEEE Computer Society Press

The Institute of Electrical and Electronics Engineers, Inc.

Systems Network Architecture (SNA) Networks

Edwin R. Coover

IEEE Computer Society Press
Los Alamitos, California
Washington • Brussels • Tokyo

IEEE Computer Society Press Tutorial

Library of Congress Cataloging-in-Publication Data

Coover, Edwin R., 1942-
Systems network architecture / Edwin R. Coover
p. cm.
Includes bibliographical references.
ISBN 0-8186-9131-X (case)
1. SNA (computer network architecture) I. Title.
TK5105.5.C68 1991
004.6'5—dc20 91-42993
CIP

Published by the
IEEE Computer Society Press
10662 Los Vaqueros Circle
PO Box 3014
Los Alamitos, CA 90720-1264

IEEE Computer Society Press Order Number 2131
Library of Congress Number 91-42993
IEEE Catalog Number EH0346-7
ISBN-8186-9131-X (case)
ISBN 0-8186-6131-3 (microfiche)

Additional copies can be ordered from

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
PO Box 3014
Los Alamitos, CA 90720-1264

IEEE Service Center
445 Hoes Lane
PO Box 1331
Piscataway, NJ 08855-1331

IEEE Computer Society
13, avenue de l'Aquilon
B-1200 Brussels
BELGIUM

IEEE Computer Society
Ooshima Building
2-19-1 Minami-Aoyama
Minato-ku, Tokyo 107
JAPAN

Technical editor: Frederick E. Petry
Production editor: Anne Copeland
Copy editor: Phyllis Walker
Cover art: Joe Daigle
Printed in the United States of America by Braun-Brumfield, Inc.

THE INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS, INC.

Preface

The purpose of this tutorial collection is to introduce readers to data networks conforming to IBM's Systems Network Architecture (SNA). The tutorial opens with an introduction, which explains the current appeal of SNA, provides some background information, and discusses major problems and solutions concerning SNA. The introduction is followed by six chapters; each chapter provides an introductory overview of the principal issues, followed by a brief summary of each paper that is included in that chapter. The papers in each chapter are ordered in difficulty from low to high. Given the difficulty posed by SNA's extensive technical vocabulary and associated acronyms, critical terminology is reintroduced with each chapter. Following Chapter 6 are a glossary, an annotated bibliography, and an author biography.

This tutorial can be seen as a blending of an historical account of SNA's development (from 1974 onward) with a topical approach to SNA. The historical account begins with Chapters 1 and 2 ("The Beginnings: The Early 'Star' Network and SNA Terms" and "Extending the Network") and concludes with Chapter 6 ("The Future of SNA"). The topical approach is undertaken in Chapters 3 through 5 ("SNA Operations and Maintenance," "SNA and the Office," and "SNA Network Management").

In Chapter 1, the principal focus is to introduce the reader to the initial SNA network architecture and its peculiar functional distribution and terminology. This chapter focuses on the initial SNA product offering, the market economics that influenced SNA design, the manner in which the essential network components were distributed, and the specialized terminology accompanying this distribution.

Chapter 2 traces the extensions of the architecture from 1974 to 1990 and discusses the "real-world" problems that accompanied these extensions. This chapter takes the 1974 SNA offering as a starting point and details how changes in the user environment constantly prodded IBM to extend SNA capabilities. The narrative carries SNA architectural and product extensions up to 1990.

Chapter 3 attempts to provide the reader with an appreciation of the range of tasks that fall to SNA's communications and systems programmers. This chapter focuses on aspects of SNA network creation, operation, and maintenance, including such operational aspects as user/application declarations, session types, global flow control, constructing routing tables, and device control specifications.

The perspective of Chapter 4 is that of the facilities offered the single largest group of SNA users, white-collar office workers. This chapter examines the broad area of SNA hardware and software support of office systems and office environments.

In Chapter 5, the perspective returns to that of the computer room, but with the somewhat-loftier perspective of information management across the span of a network. This chapter deals with the development of effective management support structures for SNA networks: how they are effected and how they are likely to evolve.

Chapter 6 weighs the relative contributions of technological, institutional, and economic forces in predicting the future directions of SNA. This chapter examines the various challenges to SNA: overcoming its own complexity, increasing its facilities for peer-to-peer communications, accommodating the increasing demand for protocol standards, and continuing to provide leading-edge applications support.

The intended audience for this tutorial comprises individuals who require an introduction to SNA networking. The intended reader's level of education or achievement should be equivalent to that of a graduate engineer (or an individual with an equivalent technical background) or an individual whose work experience has imparted a familiarity with computer networks.

The reader can expect to obtain an understanding of the following:

- The principal components of an SNA network and what these components do;
- Where SNA came from and the capabilities it offers today;
- SNA's operational and maintenance aspects, the relative complexity involved, and in what areas one can expect amelioration;
- The extent of the networking options offered for office environments;
- The technologies employed in managing SNA networks and the limitations of these technologies; and
- The role that SNA is likely to play in providing computer-based services in the future, including how SNA will likely evolve in the process.

Edwin R. Coover
July 20, 1992

Acknowledgments

Despite modern technology, a tutorial such as this does not exactly leap into its binding. I have had lots of help in putting this collection together. The authors whose works have been reprinted are formally acknowledged in the table of contents. It is the efforts of the many others that I wish to note here.

At its conception, I would like to acknowledge the encouragement of the IEEE Computer Society's former Managing Editor, Margaret Brown. Subsequently, the society's current Editorial Director, Henry Ayling, has been very supportive.

On the bibliographical front, and later in running down errant references, Beth Roth, Rob Joachim, Ingrid Dierks, Vye Jefferson, and Sharon Lieberman, all of MITRE Corporation's Library staff, have been unfailingly helpful.

When the text was in draft, a number of colleagues at MITRE — Tom Ramsey, Jim Johnson, Rudy Zumula, and George Thomas, as well as Dave Kreipke of McDonnell Douglas — provided valuable (and frequently humorous) suggestions.

During the formal review process, Ez Nahouraii, former Editor-in-Chief of the IEEE Computer Society Press, and Fred Petry of Tulane University, Technical Editor for this project, as well as the anonymous reviewers, variously provided encouragement, useful comments, and helpful criticisms.

Finally, in the production stage, Production Editor Anne Copeland and Copy Editor Phyllis Walker did their very best.

The errors that remain are mine, all mine.

Table of Contents

Introduction: The Current Appeal of SNA

IBM's Systems Network Architecture (SNA) is the dominant commercial communications architecture in the world. In the United States, survey data indicate that the Virtual Telecommunications Access Method (VTAM) — in many respects SNA's core component — is installed at 80 percent of the Fortune 1,000 companies.[1] The pervasiveness of SNA's influence is demonstrated in the following reconstruction of a day in the life of a middle-class, "First World" office worker:

> Up in the morning and off to work, he stops for gas, paying by credit card. At work, he fires up the PC and checks for electronic mail. At midday, he makes changes to sales or inventory figures on the corporate mainframe. In the afternoon, he contacts a travel agent about next Thursday's planned trip to Boston, confirming arrangements for the flights, hotel, and rental car. Before leaving work, he files the stub showing that this week's paycheck has been electronically deposited. On the way home, he stops at the bankcard machine and the local supermarket. At home, skimming the day's mail, he notes bills from the local and long-distance telephone companies and from the electric company, notice of a paid medical claim, three direct-mail catalogs, his daughter's report card, two expiring-membership notices, two appeals from worthy causes, notices requiring a rabies inoculation for the family dog and renewal of his driver's license, and junk mail addressed to "resident." Before he has a chance sit down and relax, the phone rings; it is a telemarketer selling tax-free bonds.

No matter how morally ambivalent we may be about the general usefulness of the services portrayed in the above scenario — and the pleasure or aggravation that goes along with them — the omnipresence of these automated services characterizes our environment in the last decade of the twentieth century. Generically classified, these automated services include

- Credit cards and credit card processing,
- Electronic mail and document distribution,
- Management information/decision support,
- Airline, hotel, and rental car reservations,
- Electronic funds transfer (EFT),
- Automated teller machines (ATMs),
- Point-of-sale (POS) machines,
- Direct-mail marketing and mass mailings,
- Corporate accounting/cash management,
- Claims processing,
- State and local government administration,
- Motor vehicle administration,
- Educational administration, and
- Telemarketing.

In most cases, the above services are delivered employing SNA.

Background

SNA was announced in 1973, and the first SNA products were delivered in 1974. SNA's development reflected the cost and technology structures of the early 1970s. Computing cycles and memory, as well as communications bandwidth, were all expensive; SNA was designed to wring the most out of these high-cost resources, while providing first local — and then remote — access to mainframe-based computer facilities. SNA's original design involved a host-centered "star" network, in which communications processing was hierarchically apportioned among the host, communications processors, and terminal controllers. SNA's subsequent wide acceptance in the commercial world was due to two factors. First, IBM dominated — and continues to dominate — mainframe and high-end mass storage markets. Second, IBM has shown both responsiveness and considerable ingenuity in extending the initial architecture to meet the demands of new customer requirements and a much-changed computers-and-communications environment.

Major problems

From the viewpoint of 1991, 17 years after SNA's first products appeared, one can point to significant problems confronting SNA. For discussion purposes, these problems — or challenges — fall into three groups. The first group of problems is characterized by the accretions of time; that is, they are "old business." SNA, which has automated so many functions in our late-twentieth-century society, is badly in need of automation itself. The second group of problems concerns SNA's ability to adapt to a new environment of distributed computing resources, and particularly its ability to facilitate interoperability with non-IBM and non-SNA elements. The third group of problems has to do with supporting new application areas — for example, process automation and financial services — in which there are new levels of expectation for performance, security, and interoperability.

Solutions

None of the problem areas described above would come as a surprise to IBM or major SNA users; a number have been the focus of major efforts during the last half-dozen years. Some of the problems are very difficult; all are compounded by IBM's huge installed base and broad product line.

Relative to the first group of problems (reducing SNA's operations and maintenance load), IBM has been working very hard not only to introduce significant automation into NetView — SNA's network-management product — but also to increase the communications capability and flexibility of its new products, while simultaneously striving to reduce the number of product-specialized session types.

Relative to the second group of problems (supporting a much-changed computing environment and one in which both economics and management considerations argue for a physically distributed, and often a mixed-vendor, environment), IBM has already taken important steps. After a slow start, IBM now fields credible hardware and software contenders in the local area network (LAN) market. Most of IBM's newest products offer its Advanced Program-to-Program Communications (APPC) capability, allowing program-to-program applications support. Its premier small-systems product, the AS/400, can be flexibly networked in largely autonomous, self-managing configurations via the Advanced Peer-to-Peer Networking (APPN) offering. Through Systems Application Architecture (SAA), a software-standardizing initiative, IBM has started to put its own application house in order. Included in SAA are a number of important interface specifications for SNA. Finally, IBM has moved to support well-known industry standards, such as Ethernet and the Department of Defense's (DoD's) Transmission-Control Protocol/Internet Protocol (TCP/IP). Leveraging its huge corporate resources, IBM leads other Original Equipment Manufacturers (OEMs) in delivering a full, seven-layer implementation of the Open Systems Interconnection's (OSI's) protocols, as well as the Government Open Systems Interconnection Profile (GOSIP) subset.

Relative to the third group of problems (extending SNA to meet the challenges of the new applications areas), IBM has responded with a veritable slew of hardware, software, and communications products. For the highly available, on-line transaction processing (OLTP) markets, IBM has recently introduced two products: the fault-tolerant architectured System 88 (a repackaged Status/32 Continuous Processing System) and multiprocessor versions of its System 370 (S/370) architecture (now called Enterprise System/9000) with the ability to rapidly switch applications between processors through the Extended Recovery Facility (XRF). For the broad, horizontally communicating "process control" markets — which subsume applications as various as managing POS-based supermarket systems, automating paperwork flows, and controlling industrial robots — IBM has fielded essentially three sets of products: the Unix-operating-system-based RS/6000, the OS/2-based PS/2, and the OS/400-based AS/400. Each of these three sets of products is aimed at different, but ultimately overlapping, market sectors.

In support of the above two application areas or markets (that is, OLTP and process control), IBM has provided complementary improvements. In software, for IBM's flagship Multiple Virtual Storage (MVS) operating system, addressing extensions (Extended Address) have been performed to accommodate larger applications. To boost throughput, huge memory caches and the option of substituting large amounts of extended memory for Direct Access Storage Devices (DASDs) with its Enterprise System Architecture (ESA) are now options.

In applications software support, IBM has recently introduced a series of performance-enhancing facilities for its Information Management System (IMS) and DB2 database-management systems, and its Customer Information Control System (CICS) transaction-processing application system.

In communications products, to support OLTP IBM-produced dual-processor versions of the 3745 Communications Controller, IBM began to comarket the Network Equipment Technologies (NET) line of smart multiplexors and delivered the channel-attached 3172 Interconnect Controller to support high-performance LANs. The 3172 Interconnect Controller represents a significant step in that an IBM mainframe, or mainframe complex, can function as a "server" to workstation-based "clients," communicate over several varieties of standardized high-speed local networks, and coexist with non-IBM processors. Likewise, IBM's three low-end product offerings (the RS/6000, the PS/2, and the AS/400) offer several connectivity options: Token Ring, Ethernet, Token Bus, and SNA Logical Units (LU) 2 (terminal to host) or 6.2 (application to application). Further, there is a similar alphabet soup of multiple choices in upper-level protocols: SNA, DoD's TCP/IP, International Consultative Committee for Telegraphy and Telephony's (CCITT's) X.25, General Motor's Manufacturing Automation Protocol (MAP), and the OSI and GOSIP protocol sets. Finally, with new applications, standards groups, and the deregulated market all influencing SNA development, what is quite complicated now — the historical accretion of SNA developments from 1974 through 1990 — will soon be even more complicated.

References

1. Computer Intelligence, La Jolla, Calif., cited in "Trends," *Computerworld*, Aug. 13, 1990, p. 102.

Chapter 1: The Beginnings: The Early "Star" Network and SNA Terms

IBM's Systems Network Architecture (SNA) is the dominant architecture for data networks worldwide. Whether measured in bits, bytes, packets, frames, hosts, network devices, or connect hours, probably more data travel over SNA networks than over all other data networks combined. SNA has attained this commanding position for three principal reasons. First, IBM and IBM-compatible architectures dominate both the mainframe and the personal computer segments of the world market; typically, SNA networks link these two segments. Second, as a large and well-documented "open" architecture, SNA has attracted a large market of secondary suppliers; today IBM's Virtual Telecommunications Access Method (VTAM) is the only principal SNA component without alternate suppliers. Third, since 1974, SNA has steadily evolved to meet changing customer needs and has grown from a single, host-centered "star" network to huge, multihost, multidomain networks incorporating LANs, private branch exchanges (PBXs), public data networks (PDNs), support of non-IBM computer architectures, peer-to-peer relationships, and OSI protocols.

History

SNA's origins were a far cry from being the dominant communications architecture emanating from the dominant computer vendor. SNA began as an attempt for IBM to get its own house in order, and only secondarily as a means to manipulate the larger market. A data center witticism runs,

> Question: How could God create the heaven and the earth in six days?
> Answer: He didn't have to deal with an installed base!

By the early 1970s IBM's disparate product lines employed a very mixed bag of communications access methods, protocols, and terminal types. Internally, communications methods were proliferating combinatorially. One analyst wrote that prior to SNA, IBM had more than 200 communications products requiring 35 teleprocessing access methods and 15 different data link procedures. Most of all, customers were demanding remote access to mainframe computer systems. The apparent solution, announced in 1973, with the first products appearing the following year, was essentially a company standard, somewhat grandly called "Systems Network Architecture," as it provided a blueprint for protocols, applications interfaces, device types, and product capabilities.

SNA was foremost a product of its time. It unabashedly presumed that remote users, with their terminal and remote job entry stations, wanted to access the host mainframe. Its structure faithfully reflected the economic realities of the day: that memory and processing were expensive; that analog leased lines from AT&T, then the government-regulated telecommunications monopoly, were noisy and expensive; and that maximum use must be wrung from these expensive facilities.

SNA was not significantly different from the offerings of its contemporaries — the infamous "BUNCH" (Burroughs, Univac, NCR, Control Data Corporation, and Honeywell) and the upstart minicomputer makers (Digital Equipment Corporation [DEC], Hewlett-Packard [HP], and Data General Corporation). Most of the intelligence was in the host, with a minicomputer handling what was called "hub polling." For sites with a number of devices, IBM (like Burroughs) employed cluster controllers, which polled attached devices and created a "two-level" or "hierarchical" polled environment.

Network addresses, initially of 16 bits, reflected this hierarchical environment. The first part of the address specified the subarea addresses (or the number of peripheral devices attached to a particular host or communications controller) and could be from two to eight bits; the second part of the address specified the element addresses (or the number of potential sessions allocated to terminal devices) and could be from eight to 14 bits. Although the 16 bits could be divided to suit the customer, the split — called the "address split" — had to be uniform across the network. In 1985, Extended Network Addressing (ENA) increased the total address to 23 bits, setting subarea addresses at eight bits (for a maximum of 256 subareas) and element addresses at 15 bits (for a maximum of 32,768 elements). In 1988, ENA itself was extended: The total address was increased to 31 bits and the subarea address was doubled to 16 bits (yielding 65,535 subareas), while the element address stayed at 15 bits (for 32,768 elements).

As the recurring cost of leased circuits represented the preponderant network cost, remote devices were connected by single, low-speed circuits emanating out from the host in what was metaphorically called a "star net" or a "hub-and-spoke" arrangement. Given the high costs and low speeds of the predivestiture communications environment, groups of remote terminals were typically "multidropped," where a remote cluster controller might poll terminals on line segments zigzagging into several contiguous states.

The reader may wonder why so much apparent effort is spent on what may be characterized in retrospect as the "bad old days" of data communications. The reasons are several. The first is competitive. Burroughs and Univac have merged into Unisys, but the new entity has lost market share in the mainframe market. Control Data Corporation and Honeywell are shadows of their old selves. But IBM flourishes, and SNA with it, and only DEC also has a proprietary communications architecture with a considerable following. A second reason is that today there are still many IBM SNA networks that strongly resemble the network conception first delivered in 1974. Automatic teller machines may have replaced the old terminals, punched cards, and remote job entry (RJE) stations, but many institutions still run a few bread-and-butter applications, terminal to host, year in and year out. A third reason — perhaps the most important one — is the environment from which SNA came. One must understand this environment in order to appreciate the way SNA has evolved, as well as the future directions in which it is moving.

SNA terminology

Understanding SNA terminology is one of the major hurdles to understanding SNA. Much of this terminology appears to be arbitrary, established either to thwart efforts to describe aspects of the network architecture in more generic or readily understandable terms or simply to render the familiar vaguely honorific. For example, a network-attached 3270 terminal may be described as a "Physical Unit 1 (PU 1)," a 3270 port as a "Logical Unit 2 (LU 2)," and so on. Sometimes familiar terms are used in ways that are clearly confusing, such as when a store-and-forward utility (SNA Distribution Services [SNADS]) is described as providing "asynchronous" delivery service. Finally, "SNAers" often speak in a shorthand that combines IBM hardware and software products, SNA terms, and network slang, such that their speech is unintelligible to outsiders. For example, they might say, "We are going to have to regen the TGs on the 3725s when we bring up the Type 2.1 nodes." The meaning of this sentence is clear to the "insider": "Those parts of the routing tables in the Network Control Programs (NCPs) of the 3725 Communications Controllers that deal with primary and alternate routes (the transmission groups [TGs]) will have to be changed to accommodate the presence of several new distributed computer systems (typically AS/400s) that operate as Physical Type 2.1 or Low Entry Network (LEN) sites."

In this tutorial, every effort is made to first explain SNA in generic terms and then to provide the SNA terminology (along with its inevitable acronym). Even so, first-time acquaintance with the voluminous and arcane SNA terminology will present a major stumbling block to the reader's understanding. To attempt to mitigate the terminology problem, the author has provided a glossary at the end of the tutorial and has followed the convention of purposely reintroducing the most important of the terms and acronyms from chapter to chapter, with the hope of aiding in their retention.

SNA protocol layers

In the manner of most modern communications protocols, SNA consists of a number of separate layers. Typically, layers communicate with the layers above and below by means of information headers. These headers are stripped off as the message ascends the protocol stack and approaches the user. Contrariwise, as the user's message is dispatched to distant parts, headers — like so many shipping documents — are added at each level. At the price of processing overhead, the protocol layering provides modularity and accommodates horizontal changes within the protocol stack. To SNA's credit, its designers were among the first to realize that — as the result of specialized input-output processors — the protocol "overhead" would seldom translate into processing delays. Indeed, within modern and increasingly distributed computer systems, these communications functions (that is, protocol processing) could be performed largely in parallel, resulting in a form of massive parallelism at the network level.

To those familiar with the popular "Reference Model of Open Systems Interconnection," OSI and SNA layering have apparent similarities; however, they do differ significantly (see Table 1).

Table 1. A comparison of OSI and SNA layering.

Layer	OSI	SNA	Units
7	Application	Unspecified	Useful work
6	Presentation	Transaction services Presentation services	Syntax
5	Session	Data flow control	Dialogue
4	Transport	Transmission control	Messages
3	Network	Path Control	Packets
2	Data link	Data link control	Frames
1	Physical	Unspecified	Bits

SNA preceded the OSI Reference Model as an intellectual conception by a good 10 years. SNA's layering came as a conscious and well-thought-out response to the entropy of IBM's pre-SNA situation. There is little exaggeration in observing that SNA was built and then specified, as opposed to OSI being specified and then built. SNA layering was a conscious choice of complexity over chaos. Symptomatic of the larger, functional differences between OSI and SNA are the facts that the layer names are different and that SNA did not explicitly specify the applications and physical layers. When one compares OSI and SNA layer by layer, one finds considerable differences in which functions are done where. Adding to the confusion, IBM documents have at different times described the layers differently. In addition, because SNA had to accommodate fielded software and hardware products, SNA is more complex and idiosyncratic than OSI. For instance, SNA header types are extremely numerous. (Those who anticipate having to do protocol or network analysis would be well advised to study the relevant chapters in Cypser,[1] which covers these aspects of SNA in considerable detail.) Finally, SNA predecessor products — such as the Customer Information Control System (CICS) and the Information Management System (IMS) — contain embedded communications functions that are often referred to politely as "unarchitectured SNA."

SNA in 1974: Virtual Telecommunications Access Method (VTAM), 37XX Communications Controller and Network Control Program (37XX/NCP), Synchronous Data Link Control (SDLC), and the 3270/SDLC-terminal family

Despite IBM's great resources, early SNA was an extremely modest affair. It consisted of a new communications access method, a front-end processor and software, a synchronous line protocol, and a line of modified products (most significantly, the 3270-terminal equipment line) that would use the new line protocol. Of far greater importance than the initial products was the promise — a promise on which IBM delivered — that future products, as well as selected current products, would be able to communicate with each other using SNA. Discussed below are the components of early SNA: the Virtual Telecommunications Access Method (VTAM); the 37XX Communications Controller and Network Control Program (NCP); the Synchronous Data Link Control (SDLC); and the 3270/SDLC-terminal family.

Virtual Telecommunications Access Method (VTAM). In SNA jargon, a "session" is created when an application logical unit (LU) achieves a Bind with a physical device's corresponding LU. Each physical device, such as a terminal or a printer, has an associated physical unit (PU). VTAM provides the software "switchboard" in which the LU of the physical devices connects with the corresponding LU of an application or application subsystem. A typical case would be a 3270 terminal (LU Type 2) attaching to VTAM to query an account database application or connecting to the CICS applications subsystem to do data entry. Aside from acting as connection manager, VTAM also controls the buffer allocation for the data flows back and forth between the application and the front-end processor in what SNA calls "global pacing." This does not seem to be the stuff of revolutionary change but, for IBM, VTAM allowed a number of important improvements over previous access methods. First, it standardized and modularized the formats of physical-type/logical-type access on the user side and of applications/applications subsystem on the host side.

Second, and consistent with the then-current costs of memory and processing cycles, VTAM created the initial set of the SNA hierarchy of device and session types (see Table 2).

Table 2. The initial set of the SNA hierarchy of device and session types.

Role	Device type	Session type
Host mainframe	Physical Type 5	
Communications controller	Physical Type 4	
Cluster controller	Physical Type 2	
3270 terminal (BSC)	Physical Type 1	Logical Type 0
3270 printer	Physical Type 1	Logical Type 1
3270 terminal	Physical Type 1	Logical Type 2
3270 keyboard printer	Physical Type 1	Logical Type 3
6670 Information Distributor	Physical Type 1	Logical Type 4
(S/3X) 5250 terminal	Physical Type 1	Logical Type 7

BSC = Bisynchronous, an earlier kind of terminal control protocol

Third, VTAM popularized early applications subsystems such as the Customer Information Control System (CICS) and Information Management System (IMS), whereby applications designers could achieve greater flexibility and power than before. By connecting to CICS, for instance, a valid user could run any program under CICS, rather than have his terminal be limited to a single application. Or by accessing IMS, the application designer could allow users to employ the programming economies of a database-management system (DBMS).

Fourth, VTAM continues to hold a central position within SNA. Lately (as detailed in the Chapter 6, "The Future of SNA"), one can have an SNA network without VTAM; overwhelmingly, VTAM (or VTAMs), often in conjunction with the Advanced Communications Function (ACF/VTAM), provides the focus, called the "System Services Control Point (SSCP)," of today's host-dominated SNA networks. In fact, such is the attraction of third-party suppliers (known as "plug-compatible manufacturers [PCMs]") to the huge SNA market that it is possible to have an entire SNA network in which the only IBM-supplied components are VTAM and the host operating system. Further, and covered in greater detail in Chapter 3, "SNA Operations and Maintenance," VTAM's technology, particularly the manual specification of network physical and logical units, applications, and application subsystems, has had enormous effects.

37XX Communications Controller and Network Control Program (NCP). The 37XX Communications Controller and its Network Control Program (NCP) are other core elements of SNA. The functions of the 37XX Communications Controller and its NCP program are to define the physical (wire) and logical (source-to-destination) paths of the remote connection, to poll "downstream" devices, and to buffer the traffic such that the various attached devices are "fed" the appropriate amounts. The 37XX controller family is a venerable one, with many members — including the 3704, 3705, 3725, 3710, and 3745 — and many models. Basically, the 37XX Communications Controller is a minicomputer (or, in the case of the high-end 3745s, two minicomputers), mounted in vertical cabinets, that controls a large number of speed- and protocol-specific line cards. It has long been a highly profitable product for IBM and has spawned longtime plug-compatible competitors in the form of NCR's (now AT&T's) Comten Division and the Amdahl Corporation.

As noted earlier, IBM was not alone in discovering the utility of off-loading part of the teleprocessing cycles onto a front-end processor (FEP). However, IBM was unique in the way it split the teleprocessing workload between host mainframe (an area of longtime IBM dominance) and FEP. Although IBM's motives and wisdom concerning the nature of this split can both be questioned, the split is a fact of life in

SNA and SNA networks. An artifact of this functional split, in which the host-based VTAM manages the port-to-application connection and the FEP handles the route selection and line control aspects, represents the technology by which the split is effected. The NCP (usually called the "Advanced Communications Function/Network Control Program [ACF/NCP]" in its later versions) is largely a manually specified mapping of terminals, lines, protocols, buffers, and routes that must be generated on a 370-architectured mainframe and downloaded to the FEP.

Synchronous Data Link Control (SDLC). One of the most sagacious aspects of the 1973 announcement of SNA was the decision to employ the Synchronous Data Link Control (SDLC) protocol for the link level (OSI Layer 2). The functions of the data link protocol are to package the transmitted bits and to safely deliver them across the transmission links. In selecting SDLC over the popular asynchronous and bisynchronous alternatives, IBM positioned itself well for a number of subsequent developments.

First, the new SNA protocol standard would be synchronous. Even if used in a half-duplex (that is, one-way or one-way-alternating) mode, synchronous protocols, because of less protocol overhead, typically allowed for greater transmission efficiencies than asynchronous protocols. At the time, this was reflective of IBM's concern both for line costs and for the fact that a large portion of network traffic was in batch transmissions to and from RJE stations.

Second, like the bisynchronous protocol popularized with System 360 and the 3270-terminal family, SDLC employed a 16-bit cyclical redundancy check (CRC), providing a much more powerful error-checking capability than the parity checking used with asynchronous protocols. The choice of CRC reflected IBM's concerns both with the quality of the analog circuits widely available and with the importance of data integrity for many of its customers, especially those in the banking industry.

Third, unlike either the asynchronous or the popular 3270 bisynchronous protocol, SDLC was a modern, "bit-oriented" protocol with a view to the future. Its binary codes were unrelated to any particular character set and were divorced from terminal control functions. With SDLC, the application defines the codes, and this allowed IBM much greater design latitude in defining foreign-language character sets, in transporting graphics or image data, and in similar operations.

Fourth, and also distinct from the competitors of its day, SDLC could operate in full-duplex mode, in which each station transmitted simultaneously and acknowledgements for traffic received from one direction could be "piggybacked" onto data sent in the opposite direction. Although full-duplex transmission was seldom used in early SNA implementations, its advantages became apparent once many sessions were multiplexed on high-bandwidth media and once high-volume host-to-host transfers took place.

A final SDLC feature that positioned SNA well for the future was that it could be modified — and it later was — to accommodate a greater number of unacknowledged or "outstanding" SDLC frames. With bisynchronous transmission, even though the block size or length of the transmission could be quite large, each block had to be acknowledged after each transmission, creating a "stop-and-wait" inefficiency. With SDLC, on a good-quality line, one could employ "modulo 7," or send seven frames before requiring an acknowledgement. Later, when satellite circuits, which combined the advantages of high bandwidth and low error rate with the disadvantage of a 0.5-second round-trip delay, became available, SDLC was modified to allow "modulo 128." This allowed the transmission of up to 128 frames without requiring an acknowledgement, thus allowing efficient use of this high-bandwidth medium.

The 3270/SDLC-terminal family. Although SDLC and its benefits were extended to other terminal families associated with smaller, specialized processors — such as those used with IBM Systems 34, 36, 38, 4700, 5500, and 8100 — the 3270-terminal family has been by far the most popular type. Of all the terminals worldwide, 25 percent are estimated to be of the 3270 type.

Although the 3270-terminal family, along with the System 360, was introduced in the 1960s using a bisynchronous protocol, SDLC-capable terminal controllers were introduced in the 1970s, and SDLC slowly became the standard. More significantly, the 3270 terminal's "style" — or what would now be called its "look and feel" — became the standard in many industries. The 3270 terminal functioned in "block mode"; that is, it communicated in what the user saw as screens (which actually concealed a complex data stream), at a time when much of the data processing industry was still line and character oriented. Also, it featured "function keys," whereby oft-used commands or command sequences could be stored and activated by the use of a single key. As memory prices fell, the old Control Unit Terminal (CUT) 3270s were replaced by the Distributed Function Terminal (DFT)-mode 3270s. The newer 3270 DFT terminals allowed local storage of screen templates, "hot keying" between applications (thus, multiple "LU" appearances to VTAM), document scrolling, and a variety of other innovations aimed at greater efficiency in data input and output.

Relative to SNA, the strong reception of the 3270-terminal series had a number of obvious and not-so-obvious influences. First, as the 1970s saw the spawning of vast terminal networks in the form of reservations systems and on-line services, there were heavy pressure and large constituencies demanding high availability and fast response times. This pressure tended to "lock in" conservative practices in the SNA world at the same time that others were innovating with newer methods. Second, SNA communications, responding to the same pressure, became skewed to accommodate huge numbers of half-duplex and asymmetric, terminal-to-host sessions. The asymmetry was created by the widespread practice of storing input screens locally to transmit as few as possible characters "up" to the host. But on the return trip ("down" from the host), applications-stored screens were often voluminous. Finally, and most significantly, the 3270-terminal legacy was passed on to the 1980s and 1990s in the form of 3270 emulators for the PC and what is now the Personal System/2 (PS/2) line. Although the mechanics of achieving 3270 emulation was an area of considerable innovation, 3270 emulation became the norm in communicating with the corporate mainframe. As a result, although newer, far more capable devices are typically in user's hands, the nature of the user interaction with mainframe-based applications often remains unchanged.

Papers included in this chapter

The papers that are included in this chapter are ordered chronologically and by difficulty. The first paper, "IBM's Systems Network Architecture," by IBMers Gray and Blair, appeared in *Datamation* in 1975. This paper, which was intended to present SNA to the data processing community, is of interest at several levels. First, it is clear that irrespective of the features and efficiencies of SNA, a main purpose of the SNA announcement was to put IBM's own, rather disorderly telecommunications house in order. Second, the paper goes to great lengths not to frighten the IBM customer base; considerable care is taken to describe how one can migrate to SNA step-by-step. Third, with this paper, SNA neophytes get their first small dose of SNA terminology.

The second paper, "An Introduction to Network Architectures and Protocols," by Green, is reprinted from the *IBM Systems Journal*. This paper is aptly titled, as it is a general introduction to data communications networks and the functions they perform. Toward the end of the paper, Green overviews the development of SNA, minicomputer networks, and DECnet, as well as the technologies of fast circuit switching and packet switching.

The final paper, "SNA Implementation and Architecture," by Guttman, is reprinted from the British journal *Computer Systems in Science and Engineering*. In a series of diagrams, starting with pre-SNA IBM networks, Guttman traces how SNA developed in both functionality and complexity. In a methodical and straightforward manner, the paper introduces the reader to the considerable internal complexity — and extensive vocabulary — of SNA.

References

1. R.J. Cypser, *Communications Architecture for Distributed Systems*, Addison-Wesley Pub. Co., Reading, Mass., 1978.