GC30-2022-0

IBM System/360 Operating System Telecommunications Access Method(TCAM) Concepts and Facilities

Systems

Program Number 360S-CQ-548

IBM

GC30-2022-0

Systems

IBM System/360 Operating System Telecommunications Access Method(TCAM) Concepts and Facilities

Program Number 360S-CQ-548

This publication provides a basic introduction to the concepts of (and the facilities provided by) the IBM System/360 Operating System Telecommunications Access Method (TCAM). It also briefly describes the characteristics and operating concepts of a computer-based telecommunications system.

TCAM is a complete, centralized, real-time telecommunications access method that controls allocation and use of all system resources. TCAM facilities control the transfer, editing, and processing of data from remote stations. Variety, flexibility, and modularity of these TCAM facilities permit selection of the necessary support for any telecommunications application.

A basic knowledge of System/360 machine concepts and Operating System data management techniques is required for understanding this publication.

IBM

Preface

This publication is designed to serve the informational needs of readers who may be involved in the planning, installation, or use of a teleprocessing system using TCAM as its access method.

A general knowledge of System/360 machine concepts, the System/360 Operating System using MFT-II or MVT, and Operating System data management techniques is required for understanding this publication.

TCAM is a generalized input/output control system that extends the techniques of data management to a teleprocessing environment. In addition, TCAM provides a high-level, flexible, message control language that can be used to control message traffic between remote stations, and between remote stations and application programs.

The subject matter in this publication is divided into nine major sections, plus an index, as follows:

- The INTRODUCTION, which briefly describes the general functions and capabilities of TCAM.
- TELEPROCESSING SYSTEMS, which describes the characteristics and concepts of a computer-based teleprocessing system: what it is, how its parts are connected, how communication proceeds, and how control is maintained.
- TCAM CONCEPTS AND FACILITIES, which describes the operating environment, additional TCAM concepts, and the primary capabilities of the TCAM macro instructions; provides information about data set definition and TCAM messages and buffers; illustrates message flow within a TCAM system; provides suggestions for maintaining orderly message flow; and outlines some Message Handler functions.
- APPLICATION PROGRAMS, which describes how messages are routed to application programs, the compatibility of QTAM application programs with the TCAM system, data handling considerations in application programs, the Message Control Program interface, and the sequential access method (SAM) interface.
- TCAM SERVICE FACILITIES, which describes the features and use of the Operator Control facility, the TCAM I/O Error Recording facility, debugging aids, the Alternate Destination capability, the Network Reconfiguration capability, the TCAM Checkpoint/Restart facility, and the On-Line Test function.
- TCAM APPLICATIONS, which describes the techniques of TCAM message control and processing, and suggests several TCAM applications.
- APPENDIX A, which defines the machine and device requirements for TCAM.
- APPENDIX B, which provides a graphic description of the devices supported by TCAM.
- The GLOSSARY, which defines many terms used in this publication.

References to other IBM publications direct the reader to detailed information beyond the scope of this publication. The publications referred to are:

General Information—Binary Synchronous Communications, Order No. GA27-3004

IBM System/360 Operating System TCAM Programmer's Guide and Reference Manual, Order No. GC30-2024

IBM System/360 Operating System Queued Telecommunications Access Method—Message Processing Program Services, Order No. GC30-2003

IBM System/360 Operating System Supervisor and Data Management Macro Instructions, Order No. GC28-6647

IBM System/360 Operating System Utilities, Order No. GC28-6586

IBM System/360 Operating System Advanced Checkpoint/Restart Planning Guide, Order No. GC28-6708

First Edition (January 1971)

This publication corresponds to Release 20.0 and to all subsequent versions until otherwise indicated in new editions or Technical Newsletters. Changes are periodically made to the information herein; before using this publication with IBM systems or equipment, refer to the latest SRL Newsletter for the editions that are applicable and current.

Requests for copies of IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

This manual has been prepared by the IBM Systems Development Division, Publications Center, Department E01, P. O. Box 12275, Research Triangle Park, North Carolina 27709. A form is provided at the back of this publication for reader's comments. If the form has been removed, comments may be addressed to the above address.

Contents

Illustrations

The IBM System/360 Operating System divides requests for input/output (I/O) operations on data sets into two groups, those involving either queued or basic access. Each technique is identified according to its treatment of buffering and of I/O synchronization with processing. The combination of an access technique and a given data set organization (for instance, sequential, partitioned, indexed sequential, and direct) is called an *access method*. Among these access methods is the Telecommunications Access Method (TCAM), a queued level access method using GET/PUT facilities, which controls data transfer between main storage and remote stations.

TCAM is a generalized I/O control system that extends the techniques of data management to the teleprocessing environment. Program data sets used by the problem program are queues of messages coming in from, or going out to, remote stations via communication lines. Although the time and order of the arrival and departure of messages to and from the central processing unit (CPU) are unpredictable, the program can handle messages as if they were ordered sequentially.

TCAM, however, furnishes far more than the control for I/O operations. In addition to controlling the transfer of messages between TCAM and user-written application programs, TCAM provides a high-level, flexible, message control language. TCAM macro instructions can be used to construct an installation-oriented Message Control Program (MCP) that controls the flow of message traffic from one remote station to another (message switching application), and between remote stations and any application programs (message processing applications).

A TCAM Message Control Program (MCP) is generated from assembler macro instructions coded by a user programmer. Although the assembler macro generator is used, the process folowed is similar to that used by a high-level compiler. A TCAM MCP is open-ended in the sense that the user can include functions not provided by TCAM by employing operating system control program macro instructions, and assembler language instructions and macro instructions. For instance, various TCAM macros can return either an address or a value in specific registers to which the user has access; the user can, in turn, use this information to update special counters, change various fields in main storage, and specify various functions in the message handling section of his MCP that normally would not act on the message being processed.

An MCP is completely device-dependent, with all communication lines and terminals identified to the system. Through data set definition and control information macro instructions, the user specifies his equipment configuration and the main storage areas (buffers) required for message processing in his applications. These macros generate the tables and lists of control information that define the environment of the system. An important resource in a teleprocessing system is the buffers, the number and size of which are specified by the user. The buffers are allocated from a single buffer pool as they are needed during execution.

MCP functional macro instructions provide TCAM logic modules for many procedural functions such as message routing, message editing, and error checking. By selecting the appropriate macro instructions and their operands, the user specifies the TCAM logic modules to be incorporated into his MCP and the actions they are to perform. In this way, the user tailors the system to the exact requirements of the applications being supported.

A teleprocessing control system created through the use of the TCAM message control language can:

- establish contact and control message traffic between a central computer and local or remote stations,
- delete and insert line control characters automatically, thus freeing the user from line control considerations,
- assign, use, and free buffers as required during execution (dynamic buffering),
- edit incoming and outgoing messages (for example, code translation, insertion or deletion of fields in message headers),
- forward messages to destination stations and application programs,
- take corrective action and provide special handling for messages containing errors, and
- maintain statistical information about message traffic.

TCAM services that support application programs enable a programmer to process messages from a teleprocessing network with the same easy-to-use macro instructions that he employs for sequential data handling with his local I/O devices. Because a TCAM MCP is used to perform the I/O operations, an application program may be device-independent; however, in order to correctly structure that portion of the program that governs message transmission, the programmer must have a general understanding of line control concepts. The application programmer may be, in effect, completely shielded from the time and device-dependent aspects of the teleprocessing environment. However, facilities may be provided in application programs to influence operations in the MCP. By using some other access method for a sequentially organized file, the user can write and test basic functions of his application programs without running in a teleprocessing environment. (For example, test input from a card reader can be used for this purpose.) Then, by simply changing the DD job control statement that defines a data control block (DCB), he can execute the application program under TCAM control.

This section describes the characteristics and operating concepts of a computer-based teleprocessing system: what it is, how its parts are connected, how communication proceeds, and how control is maintained. Some terms that are used throughout the publication are defined in this section (see the glossary for further definition of these and other terms.) Since this section is intended to explain teleprocessing systems to the TCAM user, it does not attempt to describe all kinds of teleprocessing systems.

A teleprocessing system (or network) consists of:

- a central computer,
- remote stations, and
- communication paths (lines or data links) that connect the remote stations to the computer (see Figure 1).

The central computer consists of the central processing unit (CPU) and the equipment (the transmission control unit and line adapters) by which the central processing unit (CPU) is connected to the communication lines.

A *remote station* is either a terminal or another computer. For example, TCAM supports the IBM System/360 and the IBM 1130 Computing System as remote stations. Reference to *the computer* means the CPU in which TCAM is running.

A *terminal* consists of a control unit (not to be confused with a transmission control unit) and one or more input and output devices, each of which is called a *component* of that terminal. Each input device and each output device is a separate component. A terminal enters keyed, punched, paper or magnetic tape data as input to the computer, and produces printed, punched, audio, magnetic tape, or visually-displayed data as output accepted from the computer. All messages from one terminal to another must pass through the computer; in addition, the computer may originate messages for the terminals.

Remote stations require common-carrier or user-owned transmission facilities (communication lines) in order to communicate with the computer. A *remote* station is connected to the computer through a transmission control unit (TCU). A *local* station is connected directly to a computer data channel by a local cable. All terminals are classified as remote except for the IBM 2260-2848 Display Complex (Local). This publication is devoted primarily to discussion of systems that use remote terminals.

Communication Lines

A *nonswitched* communication line connects the stations on the line for either continuous or regularly recurring periods. The computer can, under certain conditions, send messages to more than one station on the same line at the same time. A nonswitched line is called *point-to-point* if it connects a single remote station to the computer, or *multipoint* if it connects more than one station.

A *switched* communication line is one on which the connection between the computer and a remote station is established by dialing. As in a telephone network, the actual path for a given transmission is not fixed, but is automatically selected from a number of possible paths by common-carrier switching equipment. A switched line is always point-to-point, since the computer communicates with only one remote station on a line during any call.

Switched line connections are established by manual dialing, or by automatic dialing under program control. Because of this, switched lines are also known as dial lines. The dialing operation may be performed either at the central computer or at the remote station, and the called station may answer either manually or automatically (however, a dialing operation that originates at the central computer must be automatic dialing under program control). Not all of these options are available for all types of line configurations and remote stations: within the limitations imposed by equipment or programming, the user chooses among them based on his application requirements. For example, if the application involves collection of batched data from a number of remote stations after normal working hours, it would be appropriate to have each station equipped with an automatic answering facility and the central computer equipped with the automatic calling (Auto Call) facility. This would allow the application program to call the unattended stations automatically and to receive the batched data.

Each remote station on a switched line is continuously connected to the switching center (exchange) by an *access line*. A telephone number is associated with the access line. Similarly, each transmission control unit at the computer is connected to the exchange by one or more access lines. A transmission control unit usually has several access lines, each with its own telephone number; these multiple access lines permit simultaneous communication with several remote stations.

Audio Lines

The term *audio communication line*, or *audio line*, describes a switched line attached to an audio response unit (ARU) such as the IBM 7770 Audio Response Unit Model 3, and always refers to a telephone circuit. When audio response units are mentioned, it is assumed that they are attached to audio lines.

An *audio terminal* is a device associated with an audio response unit (ARU). The user keys in data at the terminal as input to the computer; an audible response is returned from the ARU. When an ARU has established a connection through an audio communication line, the computer receives the input message and sends the audio output message to the same audio terminal. Audio response units also function as control units.

Data Sets (Modems)

Each remote station, each transmission control unit, and each audio response unit may be connected to a communication line by a common-carrier *data set* (not to be confused with a *program* data set—see the second definition in the glossary). A terminal connected to a data channel by local cables does not require a data set. The precise function of a data set (or modem) varies with the application, but its general purpose is the same: to provide an interface between station and line. The programmer need not be concerned in any way with data sets. They are described here only to provide a complete, accurate overview of the line and station configuration.

Teleprocessing Networks

A teleprocessing system consists of either a nonswitched network, a switched network, or a combination of the two. Figure 1 shows line and station configurations of a system that includes both.

A *nonswitched* network consists of a number of nonswitched lines; the physical circuit connections determine which stations are associated with each line into the computer. A *switched* network is composed of switched lines and the user can by several means, specify which stations can communicate with the computer over each line.

An audio response unit uses a switched line, and the connection is always established from the audio terminal by dialing the telephone number of an audio communication line that has been activated (enabled) by the computer. An audio terminal can be connected to the computer on any desired line into an audio response unit by dialing the appropriate number.

A switched network is also used for common-carrier Wide Area Telephone Service (WATS). A WATS access line is permtted to place as many calls as desired, lasting as long as desired, to any of the telephones in a specified geographical zone, for either a fixed hourly rate or a flat monthly charge. WATS is a one-way originating service, requiring the subscriber to originate all calls. *Inward WATS* allows all incoming calls from the prescribed area to be charged to the receiving subscriber at a flat rate.

Some communication networks have characteristics Typical of both switched and nonswitched networks. In this publication, the term *switched network* refers to any network in which a direct physical connection between computer and station must be established by dialing before data transfer can occur. The term *nonswitched network* refers to a network in which the communication lines linking computer and stations are continuously established, thus requiring no dialing.

Transmission Techniques

A transmission technique is determined by the way in which data characters are controlled on the communication line. The techniques used by computers and terminals are start-stop and Binary Synchronous Communication (BSC).

Detailed explanations of these techniques are not given here, because the TCAM programmer need not concern himself with them except to be aware of what control characters are being used to regulate the flow of data on the communication line. TCAM supports devices using either BSC or start-stop techniques. BSC is used for high-speed data transmission between the central computer and another remote computer or high-speed terminal (for instance, the IBM 1130 Computing System or the IBM 2780 Data Transmission Terminal). Start-stop transmission is used for data transmission between the central computer and lower-speed remote terminals (for instance, the IBM 1030 Data Collection System or the basic IBM 2740 Communication Terminal).

Transmission Codes

Data can be represented on a communication line by any of several transmission codes. The code, or codes, used on a given line are determined by the kind of stations assigned to the line and the kind of adapters used to connect the line to the central computer. The programmer must be aware of the codes used on a line since he must sometimes specify the translation tables to be used by TCAM, if translation is desired.

Line Control

In a teleprocessing system, the central computer randomly receives data from remote stations during user-specified intervals. The operator at the central computer has no direct control over the remote stations. (He cannot, for example, correct a malfunctioning device at a remote station.)

With current techniques for transmitting data over long distances, errors are sometimes introduced into message data by unavoidable transient line conditions such as crosstalk and lightning strikes. A teleprocessing system must be able to detect transmission errors and, when possible, to correct them (by retransmitting the message or message block containing the errors). If the error is one that cannot be corrected, its presence must be made known to the user program so that the program can take appropriate action.

The scheme of operating procedures and control sequences by which a teleprocessing system is managed is called *line control* (or, for BSC, *data link control*). A line control scheme must consider the functional characteristics and capabilities of the equipment and communication lines making up the system, as well as the operational requirements of the system. Some specific factors concerning line control that must be considered are:

- how contact is to be established between stations,
- how a message is to be directed to a specific station on a multipoint line,
- what happens if two stations try to send at the same time,

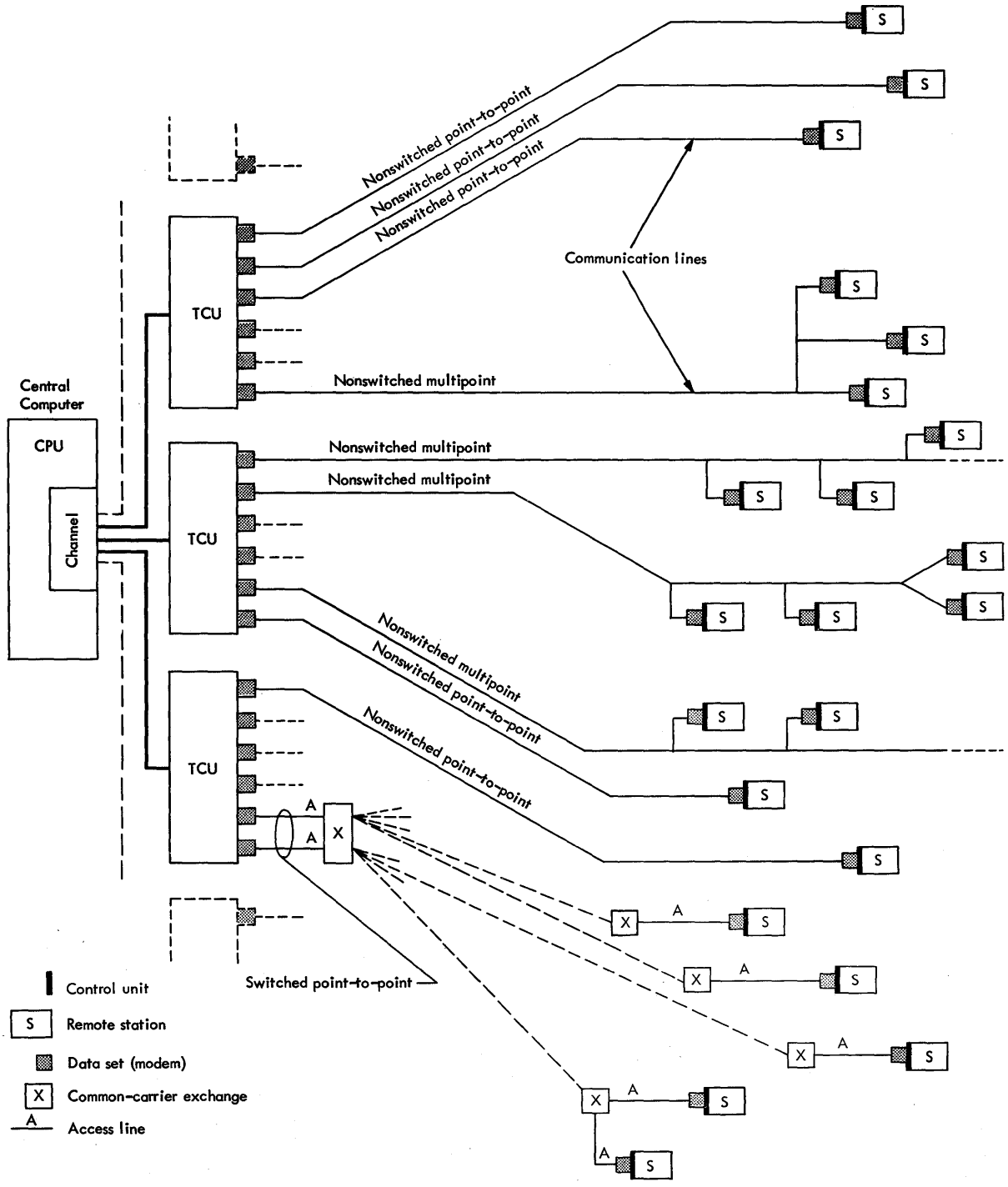

Figure 1. Line and Station Configurations

- what to do if a station fails to respond to a message, and
- how to validate transmitted message data.

Line control can be classified in two ways. The first way is by the transmission technique (start-stop or BSC) that is used for the line. With each technique is associated a set of control characters and rules for their use. Some control characters are used for both start-stop and BSC transmission, while others are peculiar to one transmission technique or the other. The second way in which line control can be classified is by the communication line configuration with which it is used. For example, line control for a switched line varies from that for a non-switched line.

While the general capabilities and functions of a given line control scheme are identified in terms of transmission technique and line configuration, individual variations arise from differences in the kinds of stations to be controlled, and by the presence or absence of certain features on the stations. For example, a given line control scheme may include the control characters needed to indicate occurrence of a transmission error and to request automatic retransmission, but some types of station equipment using that line control scheme may not be capable of error checking or automatic retransmission. Generally, all stations connected on a given line must be designed to use the same line control scheme, and a given capability must be common to all stations on the line in order to be used.

In a teleprocessing system operating under TCAM, the way messages are handled with respect to line control depends upon whether the messages are to be transmitted between similar stations (IBM 1050 to 1050, for example) or between dissimilar stations (a non-IBM terminal to an IBM 1030, for example); messages between stations and application programs are treated as messages between dissimilar stations. The distinction arises because messages between similar stations have satisfactory line control characters inherent in the incoming message, and the message can be sent without change in the line control. Dissimilar stations, however, have different line control characteristics; thus, the input message must have line control deleted as necessary, and the output message must have appropriate line control inserted. In either of these situations, the TCAM programmer needs to specify whether he wants line control to be ignored by TCAM or to be removed and reinserted in messages by TCAM.

For stations using the BSC technique, another transmission variable involves the treatment of line control characters in a message. BSC messages may be transmitted in transparent or nontransparent mode.

When transmitting in *transparent mode*, all characters in the message, except for an initial line control character designating transparent mode and except for the end-of-block signal, are entered by the station as data. When the first block of a message is transmitted in transparent mode, the remaining blocks must also be in that mode and must be transmitted in a fixed format acceptable to the receiving device.

In *nontransparent mode*, all line control characters are treated as such and line control is handled as it is for start-stop stations. Thus, some characters that may be transmitted as text in transparent mode are line control characters in nontransparent mode.

The initial line control sequence placed in the message designates the message as being in transparent or in nontransparent mode when it is entered by the station. When the message is being sent to a BSC station by TCAM, TCAM must insert the initial sequence.

Additional BSC information can be found in the SRL publication *General Information—Binary Synchronous Communications*, Order No. GA27-3004.

Invitation and Selection of Stations

This section provides a general description of the control of messages to and from stations connected in a teleprocessing system. Since TCAM is a computer-controlled system, only such systems are discussed.

The computer contacts a remote station either to:

- allow the station to enter a message on the line if it has one ready (called *invitation*), or
- send the station a message (called *selection*).

A station is *invited* to enter a message on the line and is *selected* to have a message sent to it.

Stations are not always ready to handle message traffic; contact between a station and the computer must be established before a message can be transmitted. The manner in which contact is made depends upon several variables, one of which is whether the station is on a switched or a non-switched line.

For switched lines, a preliminary step is necessary before a remote station can enter a message on a line. Although remote stations are permitted to call the computer at any time, the computer, in order to fulfill its control function, must either accept or ignore incoming calls. Therefore, the computer performs a function known as *enabling the line*, which is the process of conditioning the transmission control unit (or the audio response unit) to accept incoming calls on a line. TCAM determines whether lines are to be enabled at a given moment.

The station operator can dial the telephone number (or one of several numbers) of the computer. If a station calls in on an enabled line that is not in contact with another station, the line connection is completed and message transmission can occur (the response by the computer is called answering). If the line is not enabled, the calling station gets no answer and must try again later. A busy signal results when the line is enabled but is being used by another station; the calling station must try again later.

On both switched and nonswitched lines there are several methods of controlling a line for transmitting messages (for switched lines, this process occurs after the line con-

nection is made as described above); polling, addressing, and contention are common to both kinds of lines. Identification exchange is restricted to switched lines. Polling and addressing are non-contention methods of controlling a line. When more than one polled station is assigned to a switched line, a station that is calling the computer cannot uniquely identify itself at the time it makes the call; therefore, all such stations assigned to the same line must have identical polling characters.

Polling

Polling is a method by which the computer invites remote stations to enter messages on the line. (Remote stations, for this discussion, include stations on multipoint nonswitched lines and terminals on point-to-point lines.) The computer periodically contacts the remote stations and invites them to enter any messages they have ready. (*Ready* means that the remote station operator is prepared to enter data from a keyboard, or that some input on cards (or magnetic or paper tape) has been placed in an input device so that data can be entered automatically when the device is activated by polling.)

Polling is accomplished by sending one or more polling characters. Typically, two characters are used; the first selects the remote station, and the second selects a specific component at that station. The identified station then sends a positive response to the computer if it has a message to send, or a negative response if it does not. The computer may poll a number of remote stations and components in turn, until one is found that has a message ready. In TCAM, polling characters are maintained in an *invitation list* for each line.

An *invitation list* must be specified for each line in the system. In defining the list for a nonswitched line, the user may enter station names as often as he wishes and in any order. Only stations on the same line can be included. If a line is used for output only, the stations in the list are not invited to enter messages on the line.

The user may specify that a delay occur before invitation (polling) on a nonswitched line is started again. This polling *delay* (to 255 seconds) is entered at the end of the list, if specified. When the delay expires, polling resumes at the beginning of the list. This delay is used to minimize unproductive polling, and to permit sending operations on the line (the polling delay may be altered by an operator command).

The user also may specify that a system interval be available. If it is activated, when current activity on all nonswitched lines is completed, the TCAM MCP begins a prespecified delay (up to 65,535 seconds) during which no output is sent and no stations are polled. When the delay expires, line traffic is started again (the duration of the system delay may be altered by an operator command).

For nonswitched lines, the polling process may be achieved entirely by programmed capability or by the Auto Poll machine feature. (Note that a remote station cannot initiate contact on a nonswitched line when polling is used to control that line.) Autopolled lines in this publication refer to lines that are normally polled under the control of the Auto Poll feature, while polled lines refer to lines that are polled under program control.

Addressing

Addressing is a method by which the computer selects remote stations to accept messages sent by the computer. (Remote stations, for this discussion, include stations on multipoint nonswitched lines and terminals on point-to-point lines.) For a contention system, the message is sent after the computer successfully "bids" for the line. For non-contention systems, the procedure depends upon whether the lines are switched or nonswitched.

When the computer has a message to send on a nonswitched line, it sends one or more *addressing*, or call-directing, characters. As in polling, two characters are commonly used; the first selects the remote station, and the second selects the component. The remote station identified by these characters enters a positive response if it is able to accept the message, or a negative response if it is not. When the remote station enters a positive response, the computer sends its message. If the remote station enters a negative response to the addressing characters sent by the computer, the user can cause TCAM to hold the message and try for a positive response later.

When the computer has a message to send on a switched line, it may automatically dial the telephone number of the remote station after the enable (if any) is terminated, or it may wait for the station to dial. The function performed by the computer or the station in this case is known as *calling*. The connection is established when either the remote station or the computer responds; addressing then takes place and the message is sent to the remote station.

Contention

Switched and nonswitched remote stations can *bid* for use of a line when the line is enabled. (Remote stations, for this discussion, include stations on point-to-point nonswitched lines, and terminals on point-to-point lines.) Either the remote station operator or the computer can initiate contact on an available line, and procedures for both computer-initiated and station-initiated contact are the same. A contention system determines whether the computer or the remote station is to transmit its message when the computer and a remote station simultaneously contend for the line.

Identification Exchange

Switched stations may identify themselves by automatically entering an *identification sequence* after a line connection is established. TCAM compares the sequence it receives against a user-defined sequence; transmission proceeds if there is a match. If the sequences do not match, TCAM disconnects the line.

For BSC transmission, the calling station usually enters an identification sequence and expects, in return, the iden-

tification sequence of the computer; this results in an ***identification exchange***. In start-stop transmission, however, the calling station may call without identifying itself, and the computer enters its identification sequence (there is no exchange of identification sequences).

If the remote station is a computer, the identification sequence is provided by its programmer, and the central computer can check the identity of the station. If the remote station is a terminal, the sequence is included as an installation feature of that terminal, but the sequence need not be checked if identification exchange is not required.

The Operating Environment

TCAM operates under control of the System/360 Operating System with MFT-II or MVT (or MVT only, if the Time Sharing Option is used). The TCAM Message Control Program (MCP) and any application programs may be executed as separate jobs. The MCP should be executed as the highest priority job (and suffers performance degradation if it is not). Application programs may be initiated in any of the remaining partitions or regions. Application programs may also be executed in the same partition or region as the control program after being invoked by the user through the system ATTACH macro instruction. If the TCAM jobs do not require the use of all available partitions (or regions), the remaining partitions may be used for batch processing jobs. Batch jobs should be initiated in lower-priority partitions or regions.

The application programs operate concurrently with the control program. The operating system supervisor controls the asynchronous operation of the various programming components of the system. This method of execution is based on:

- the completion of awaited events (such as I/O termination) and the availability of resources (for example, buffers), and
- the established priorities of the various tasks.

After being assembled, linkage-edited, and cataloged onto a job library, an MCP can be initiated and executed in any partition. However, the MCP should run with a higher priority than any application program, and the TCAM MCP can run efficiently only in the highest-priority partition or region. The MCP must be initiated before any TCAM application programs.

Depending upon the requirements of the user, a System/360 that includes teleprocessing may be either:

- dedicated to teleprocessing, or
- set up to execute non-teleprocessing jobs concurrently with the execution of teleprocessing jobs.

The system is dedicated to teleprocessing when all partitions or regions are allocated to programs performing teleprocessing functions.

At the other extreme, if no message processing is continuously required by the teleprocessing application, non-teleprocessing programs can be executing in any partition lower than that of the TCAM MCP, depending upon the amount of main storage available. An example of such a configuration under MFT-II is: normal batch processing in three partitions, concurrent peripheral operations in a fourth partition, and an MCP performing teleprocessing functions in the highest-priority partition. Such a configuration can exist when when the MCP can perform the required teleprocessing functions without an application program (for example, either a message switching, a data collection, or a standard audio answering application). If it is desired to terminate operation of the MCP from an application program (rather than by an operator command), an application program must be initiated in another partition or attached (by means of the system ATTACH macro) to perform this function.

It is possible to support a teleprocessing application that requires message processing by just one application program (in addition to the MCP); one application program can be designed to process all message types and to terminate the MCP when necessary. Alternatively, a number of application programs may be written to perform the processing needed for the various types of messages. No matter how many application programs are needed, they may be run in the same partition as the MCP (by means of the ATTACH facility), or they may be run in other available partitions or regions of the system.

General Concepts and Facilities

TCAM systematically and efficiently controls the flow of data in a computer-based teleprocessing system, and concurrently performs any required processing of data. Data enters the system in the form of messages from stations or from programs that generate messages. Data is ultimately delivered to one or more stations or to programs that process the data. Message flow into the system is random and proceeds at relatively slow speeds (due to the operating speeds of the terminals or the lines), while the messages, once delivered to the computer, can be processed at computer speeds. Because of these differences in speed, and for further reasons described below, TCAM support for the telecommunications system is divided into two categories, message control in the MCP and message processing in application programs.

The MCP consists of routines that identify the teleprocessing network to the System/360 Operating System, that establish the line control required for the various kinds of stations and modes of connection, and that control message handling and message routing to fit the user's requirements. A message control program (MCP) is required in a teleprocessing system operating under TCAM.

One or more user-written *application programs* may be required to process the contents of the messages. There may be several application programs or none, depending upon the needs of the installation.

An example of a job requiring no application program is message switching, where the sole function of the teleprocessing system is to receive messages from stations and forward them unaltered (except for such processing as the MCP may perform) to one or more other stations.

Message processing proceeds asynchronously with message control to fully utilize the differences in handling and

processing speeds. Another reason for having separate control and application programs is that while many device-dependent considerations govern the design of an MCP, they do not affect the design of an application program. All the application programmer needs to consider is the format of the data his program is to process.

The MCP serves as an intermediary between stations and between the stations and any application programs. The device-dependent input/output operations are performed by TCAM routines in the control program, based on the station and line configuration of the system as specified in the operands of TCAM macro instructions. To provide maximum efficiency, TCAM places messages in queues either in main storage (main-storage queuing) or on direct-access storage devices (DASD queuing); messages are subsequently recalled from the queues for either processing or forwarding to some destination. This procedure enables the application program or message handling functions to refer to messages rather than stations. This is accomplished from an application program using GET, PUT, READ, WRITE, CHECK, and POINT macro instructions.

The MCP itself can perform limited processing of the message; for instance, the MCP may examine the initial bytes of data in a message to determine routing information and message code translating. Certain optional processing operations are provided by TCAM as a convenience to the user. For example, the MCP can insert the time of day in messages, eliminating the need for an application program to do this. Further, the MCP can check the input messages to determine if an error message must be sent to the calling station. In this publication, *processing* always refers to the functions of an application program and not to the functions of an MCP, which are referred to as *handling*.

A teleprocessing system may include several different terminal types and both switched and nonswitched lines. One or more communication lines of the same type, over which terminals with similar characteristics can communicate with the computer, constitute a *line group*. For each line group, or for several line groups with similar characteristics, the user must specify a sequence of TCAM macro instructions to handle messages. These macro instructions call routines that provide the requisite handling of the messages; each such macro sequence is called a Message Handler (MH) and the routines are called MH routines. The MCP may contain several Message Handlers. However, to save main storage space, the Message Handlers for line groups that have considerable duplication of handling may be combined into a single MH, parts of which might be executed selectively for messages from different line groups by varying the path through the MH.

An MH, therefore, may be a group of routines that provide a means of properly routing message segments through the message-handling portion of the MCP. The incoming group of an MH handles messages arriving either from associated stations of a line group or from associated application programs. The outgoing group of the MH for a destination handles messages for the related destinations, which may be either stations or application programs. See *Message Flow Within the System* later in this section for a more detailed description of the functions of an MH.

TCAM Macro Capabilities

TCAM facilities include a comprehensive set of input/output, message control, translating, error handling, and editing routines that relieve the programmer of the detailed programming normally required for a teleprocessing system. The programmer is also relieved, as far as possible, from concern with line control and other details not dealing with message handling. Macro instructions allow the system programmer to select for his MCP at assembly time the TCAM routines that are designed to meet the exact requirements of the installation.

The primary capabilities of the teleprocessing programs that can be created through the use of TCAM macro instructions are:

- inviting stations to transmit messages;
- receiving messages from stations;
- selecting stations;
- sending messages to stations;
- enabling and disabling communication lines;
- assigning and using available buffers as required during program execution;
- performing message editing functions for incoming messages, such as translating from the transmission code of a non-EBCDIC station to extended binary coded decimal interchange code (EBCDIC), inserting time-received and date-received information in the message header, recording (logging) the message on a secondary storage medium such as magnetic tape, and inserting or removing user-specified data in the header and maintaining a count of the number of messages received from each station;
- routing messages to appropriate queues, determined by either the destination code specified in the message header, or by MH macro instructions;
- queuing messages on a direct-access storage device or in main storage;
- initiating corrective action when an error or unusual condition is detected;
- intercepting transmission of messages in error;
- canceling messages containing errors;
- rerouting messages to alternate destinations;
- transmitting error messages;
- routing messages with erroneous header information to a special queue;
- providing message data, in the work unit specified, to an application program;
- placing response messages generated by application programs on queues for subsequent transmission;
- retrieving messages already queued for transmission to stations;
- performing message editing functions for outgoing messages, such as placing time-sent and date-sent in-

formation in the header, placing an output sequence number in the header, inserting or removing user-specified data in the header, logging the outgoing message on a secondary storage device, maintaining a count of the number of messages sent to each station, and translating the message from EBCDIC code to the appropriate transmission code;
- taking periodic checkpoints of the system in which the status of the queues and of the teleprocessing network is saved on a direct-access storage device (this information can be used by a recovery facility in case of subsequent failure);
- providing operator-to-system control stations;
- keeping counts of station errors;
- providing error recovery procedures;
- deleting and inserting line control characters in messages;
- providing flexibility of message handling;
- providing priority handling of messages; and
- supporting binary synchronous communication.

Data Set Definition and Control Information

A data control block (DCB) must be defined for each data set referred to by the MCP and by any application programs. This is accomplished by means of DCB macro instructions, one of which must be provided for each of the types of TCAM data sets referred to by the MCP:

- each communication line group data set, which consists of the lines in a line group over which messages are transmitted to and from the CPU;
- each secondary storage message queues data set, which contains the destination queues for each destination and a queue for logging medium used; and
- the checkpoint data set, which consists of the records that are maintained and stored on a DASD, to be used for TCAM recovery.

Similarly, an appropriate DCB macro instruction must be provided for each message log data set used by the MCP (this data set consists of messages or message segments placed on a secondary storage device for accounting purposes).

The DCBs in a Message Control Program serve as logical connectors between the MCP and the associated line group, message queues, and message log data sets. The DCBs defined in an application program are not associated with data sets themselves. They provide control information that TCAM uses to transfer data to and from an application program. The destination queues are the principal connectors between the MCP and any application programs.

In addition to the data set definitions, the user must supply control information that the MCP uses to control both sending and receiving of messages. The control information consists of:

- the name and address of each station, along with related information such as any special distribution lists (for sending a message to more than one station), or cascade lists (for sending a message to the least busy of a list of stations);
- the name of each destination queue associated with an application program to which incoming messages are to be sent;
- an invitation list for each line that indicates the order in which the stations are to be invited to enter messages;
- the size and number of main storage buffers that are to be used for sending messages to and receiving messages from the stations (to compensate for the differences in the rates of information flow, TCAM automatically uses available buffers as they are needed);
- the names of the logging queues;
- the name of each audio communication line, together with related information, such as line group name; and
- certain information used to communicate between application programs and the MCP.

The user supplies this control information through appropriate TCAM macro instructions and their operands.

TCAM Messages

Message Formats

A message may consist of two parts, the header portion and the text portion. The *header* portion may contain control information for the message, such as:

- one or more destination codes;
- the code name of the originating station;
- the number of the message relative to the numbers of the previous messages received from that station (input sequence number);
- a message-type indicator, and
- various other control indicators.

The *text* portion of a message consists of information of concern to the party ultimately receiving the message, either a station or an application program.

Depending upon the application, messages may consist of just a header, just text, or of both header and text. A header-only message may use a message-type indicator to route the message to an application program and, possibly, to obtain a standard response. If all messages go to a unique application program, such as a file-update program, routing information in the header may be omitted.

The system programmer designs the message format; he decides what part of the message is the header and what is text. If other fields follow the destination field, the end of the header may be considered as the last character of the message on which the MH header functions act; subsequent characters constitute the text portion of the message.

Operations on the header fields are the primary function of the Message Handlers in the MCP. The length and format of the header and the information it contains depend solely on the requirements of the application and the user's preferences. The length may be a few characters or many. A header may occupy more than one buffer.

The format of the message header dictates the arrangement of the appropriate MH macros. Control characters that are used and the sequence of fields within the header must be predetermined so that the MCP can be properly coded.

The destination codes in the message header identify the stations or application programs to which the message is to be routed. The message-type indicator can be used to identify a header that is to be processed in a special manner. By inserting certain macro instructions in the MCP, the user can insert in the header such data as the date and time the message is received, the date and time it is sent, and the output sequence number.

If the headers vary in length, as when multiple destinations are used in a message switching application, a special character must be entered in the message to indicate the end of the list of destinations. If the destination field is the last field in the header, the end-of-addressing character indicates the end of the header.

Depending upon the type of work unit with which he is dealing, the user must specify appropriate characters for control purposes. The types of work units and corresponding control characters are as follows:

- A *block* is that portion of a message terminated by a line control character—EOB, ETB, ETX, or EOT. A *sub-block* is that portion of a BSC message terminated by an ITB line control character.
- A *segment* is that portion of a message contained in a single buffer, the size of which is specified by the user for each line group and application program.
- A *record* for an application program is, most often, that portion of a message terminated by the last data byte in the last message buffer or, less often, a message portion terminated by some user-specified data character.
- A *message* is that unit of data that is terminated by an EOT character or, if end-of-block checking is specified in the MH, by an ETX, ETB, or EOB character.

There are many possible variations for the format of a message header. The sample formats shown in Figures 2 and 3 are included simply for illustrative purposes; the format shown in Figure 2 could be used in a message switching application (it shows how an incoming message might look just before it arrives in main storage).

Figure 2. Sample Format for an Incoming Message

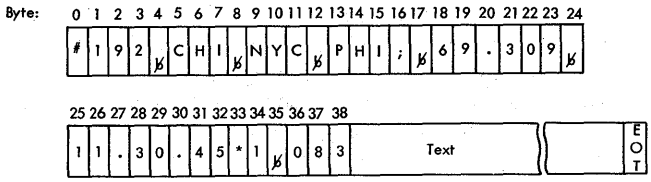

Figure 3. Sample Format for an Outgoing Message

Blanks (b) in these sample formats serve as delimiters for each header field (rather than having either a certain number or a certain sequence of characters to distinguish the limits of the fields). The first 17 bytes of each message are identical; the user specifies bytes 1 through 16 for the incoming message, and TCAM places them in corresponding positions in the outgoing message. The first byte (byte 0) contains a machine end-of-address character that is inserted by the station; bytes 1 through 3 contain the input sequence number (the total number of input messages, including this one, entered by this originating station); bytes 5 through 7 identify the originating station (the station entering this message); bytes 9 through 11, and 13 through 15, identify the destination stations for this message; byte 16 designates the program end-of-address character (user-specified). The remaining bytes of the incoming message are the user-specified priority (byte 18) and the text of the message followed by the appropriate EOT control character.

In the outgoing message, the MCP has inserted both date-received and time-received information in the header (bytes 18 through 32). The priority information has been moved (from byte 18 in the incoming message) to byte 34. Bytes 36 through 38 contain the output sequence number (the total number of messages, including this one, that have been directed to the station, or stations, designated as destination stations in the message header); the remaining bytes of text and the EOT control character in this message are identical to the corresponding entries in the incoming message.

TCAM, with its complete set of header-processing routines and associated macro instructions, allows the user to indicate the header-processing functions he wants performed by including the appropriate macro instructions in the MCP. In addition to those described briefly in this section, many other functions are available, such as the detection of incorrect or invalid information in the header fields.

Buffers

Messages entering a TCAM network are read into buffers, which are main storage areas used for handling, queuing, and transferring of message segments between all lines and queuing media and between queuing media and the application program work area. (A message segment is that portion of a message contained in one buffer.) A buffer has two parts, an area containing control information (the *buffer prefix*) and an area containing all or part of the

message. Buffers must be at least 31 bytes long, and may be no longer than 65,535 bytes.

Structure of a Buffer

To provide efficient buffering and best use of main storage, the TCAM network maintains a buffer unit pool containing buffer *units* of one size. Buffer units are the basic building blocks from which buffers are constructed.

The size and number of units in the pool are specified by the user. For internal management purposes, 12 bytes are added to the user-specified unit size. Thus, if a user specifies a unit size of 60 bytes, the size of the unit will be 72 bytes. The user should not include the extra 12 bytes when defining his buffers.

The size of a *buffer* for a line group is specified also by the user. All buffers used by a given line group are the same size, but each line group may use buffers that differ in size from those assigned to other line groups.

By linking an appropriate number of buffer units, TCAM constructs buffers containing a number of bytes at least as great as that specified by the user as his buffer size. For example, if the user specifies a unit size of 60 and a buffer size of 120, TCAM links two 60-byte units in building buffers for that line group.

There are two types of buffers, header and text. A *header buffer* contains all or any part of a message header. A *text buffer* contains message text only.

A *buffer prefix* is a control area contained within each TCAM buffer. Although the user must allow room for the buffer prefix in defining his buffers, TCAM fills in the buffer prefix area with buffer control information.

If only one buffer is used to contain a message, the buffer prefix occupies the first 30 bytes of the buffer. If more than one buffer is used to contain a message, the first buffer prefix occupies the first 30 bytes, and a 23-byte buffer prefix occupies the beginning of each subsequent buffer assigned to the message.

Thus, two kinds of control areas are associated with buffers. The 12-byte control area associated with each *buffer unit* is assigned automatically by TCAM and is not included by the user when he is defining his buffers. The 30-byte or 23-byte prefix assigned to each buffer must be considered by the user; he must allow for this area in defining the size of his units. Each unit must be large enough to contain a header prefix plus at least one byte (31 bytes) and may be no larger than 255 bytes. Note that the second and subsequent buffers contain more bytes of message than does the first buffer since their prefixes are 7 bytes shorter than that of the first buffer. Figure 4 illustrates two buffers assigned to a line group where a user specifies a unit size of 60 and a buffer size of 120 bytes.

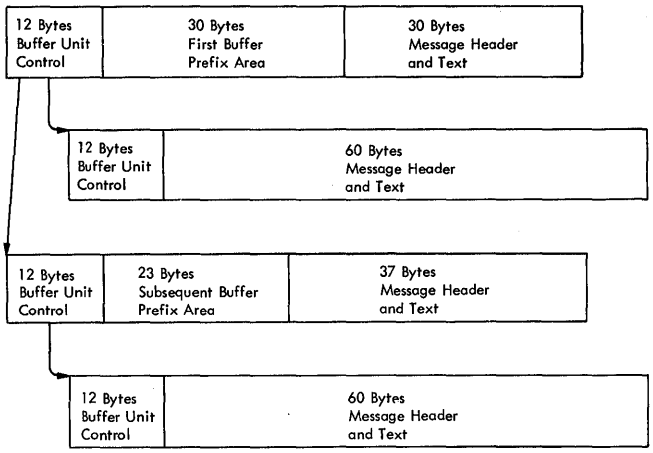

Figure 4. Buffer Format

Each buffer is composed of two linked units, and the two buffers are also linked. Each unit is 72 bytes long (the 60 bytes specified as the unit size plus a 12-byte unit control area added by TCAM). In defining the 120-byte buffer size for the line group, only the 60 bytes specified by the user need to be considered.

Remember:

- A buffer is made up of one or more buffer units.
- Each *buffer unit* must be at least 31 bytes long (a 30-byte buffer prefix plus at least one byte for data, which does not include the 12-byte control area added by TCAM) and may be no longer than 255 bytes (again, not counting the 12-byte unit control area).
- Each *buffer* must be at least 31 bytes long (minimum size of one unit) and may be no longer than 65,535 bytes.

Buffer Unit Pool

One buffer unit pool is defined for the MCP. This single pool contains a user-specified number of buffer units. The size of the pool must not exceed 65,535 units.

When message traffic is in progress, a unit in the unit pool may be in any one of three states. If a main-storage message-queues data set is specified, some units are assigned to main-storage message queues. Other units are linked to form buffers that are assigned to line groups. Finally, some units are assigned to an *available-unit* queue, where they remain until they are linked to form a buffer, or are assigned to a message queue.

Design Considerations

In designing his buffers, the user is concerned with specifying the following (provided by specifying specific operands

of the DCB macro defining each line group data set in the system):

- The number of buffers to be assigned initially to each line in the line group for receiving operations.
- The number of buffers to be assigned initially to each line in the line group for sending operations.
- The size of buffers to be used for all lines in this line group.
- The maximum number of buffers to be allocated to a line at one time.
- Whether and how program-controlled interruptions are to be used for control of dynamic buffer allocation and deallocation.
- The number of bytes (0 to 255) to be reserved in the buffer for insertion of the time-of-day, the date, the output sequence number, and any other user-specified data.

The user is also concerned with specifying the following during TCAM initialization:

- The length of a buffer unit (must be between 31 and 255 bytes inclusive).
- The number of units in the buffer unit pool that may be used to build line buffers. The total must not exceed 65,535.
- The maximum number of buffer units that may be assigned to main-storage queues at any one time.
- The number of channel program blocks (CPBs) to be provided to handle DASD queuing. A maximum of 65,535 CPBs may be specified. (CPBs are used to transfer data between buffers and direct-access storage devices.)

Management of data buffers for incoming and outgoing messages is an important factor in running a TCAM system efficiently. A system programmer must consider several factors in weighing the trade-off of time and main storage. He must:

- specify enough buffer units to ensure against loss or undue delay of data;
- select the sizes of his buffer units and buffers to accommodate his message;
- decide whether to use the program-controlled interruption (PCI) feature for control of buffer allocation and deallocation during execution (*dynamic* buffering); and
- determine the number of buffers to be assigned initially to each line in a line group for sending and receiving operations, and the maximum number of buffers to be assigned to each line.

For a more thorough discussion of buffer design and the impact of the various design factors outlined here, see the *Defining Buffers* section in *IBM System/360 Operating System TCAM Programmer's Guide and Reference Manual* Order No. GC30-2024.

Program-Controlled Interruption (PCI): A PCI may occur while either the first or a subsequent buffer assigned to a line group is being filled, provided that the appropriate keyword operand of the DCB for the line group permits program-controlled interruptions. Depending upon the keyword operand chosen, one of the following discussions on functions applies:

- When the first PCI occurs, the number of buffers assigned to the line group is equal to the difference between the maximum number assigned to a line group and the total number initially assigned to the line group. On subsequent PCIs, the buffer immediately preceding the one being filled or emptied is freed, and a new buffer is requested to keep the number of buffers assigned to the line group equal to the maximum specified by an operand of the line group DCB macro.
- The previous buffer is freed when the second and subsequent PCIs occur, but no requests are made for additional buffers.
- If program-controlled interruptions are *not* permitted, then the number of buffers assigned initially must be sufficient to handle the entire transmission (to avoid losing data), and no buffers are freed until either the transmission is completed or an end-of-block control character is received.

Initial and Maximum Assignment of Buffers: The number of buffers that should be assigned initially to each line in the line group (by means of the line group DCB macro) depends upon the following:

- terminal type;
- terminal speed;
- line speed; and
- whether PCI is specified.

The number of buffers to be assigned initially varies directly with the speed of the line and the terminal; the faster the data is transmitted, the higher the initial assignment should be. Up to 15 buffers may be assigned initially; one buffer is assigned automatically by TCAM for receiving operations and two for sending operations if the line group DCB operands that assign these buffers are omitted.

The maximum number of buffers assigned to a line group (by means of the line group DCB macro) also depends upon line and terminal speeds. For a system using PCI, program-controlled interruptions may not be accepted by the CPU in time for buffer replacement to be effective. When PCI is not used, the DCB operand that assigns a maximum number of buffers to a line group is ignored.

Specification of CPBs: Channel program blocks (CPBs) are used to transfer data between buffers and direct-access storage devices. The number of CPBs needed depends upon the amount of message traffic and the number of lines in the TCAM-controlled telecommunications system,

and upon the size of the buffer units. TCAM can operate with only one CPB, but more than one is ordinarily required for efficient operation. Specifying too many CPBs wastes main storage, since the length of each CPB is 64 bytes plus a buffer unit.

Message Flow Within the System

This section describes the flow of a single-segment message through a system operating under TCAM, from the time it is entered at the remote station to its transmission to a destination station. Figure 5 outlines the flow of a message segment through a TCAM system. In referring to a step number, the reference is to the corresponding circled number on the flow diagram. The following discussion amplifies the figure.

The header and text prefixes described in this section are generated automatically and are used by internal TCAM routines. The only programming consideration required of the user for the manipulation of the buffers and their prefixes as messages flow through the system is allocating enough main storage space for them in his buffer pool.

Steps 1 and 2: The input message is prepared at the remote station and entered on the line. The message may be keyed in, or it may be entered from a card or tape reader. The originating station enters the message via a communication line, the TCU, and the multiplexer channel.

Step 3: The message enters the CPU and is stored, together with the buffer prefix, in a main storage (primary) buffer. As message data fills the buffer, TCAM inserts chaining addresses and other control information into the prefix field. (The data in a buffer provides a convenient unit for handling; reference to handling or processing of a buffer means that portion of a message contained in the buffer.) Before the message characters are placed in the first header buffer, TCAM may reserve space in the buffer for later insertion of the time, date, and sequence number for the message, and for the screen control character for the IBM 2260 and 2265 remote display complexes. Once a buffer is filled with the first segment of the message, the MCP controls the flow of the buffer (that is, the data in the buffer) through the teleprocessing network. The heart of the MCP consists of the Message Handlers constructed by the user to process messages from the various lines or line groups.

Step 4: The incoming message is routed to the incoming group of the MH specified for the line (by an operand of the DCB macro for the line group in which the line is included). The message is passed, a buffer at a time, through the incoming group, which performs user-selected functions on the message header such as origin checking, insertion of the time, and input sequence-number checking. Similarly, functions may be performed for the message segment, such as translating the segment from line code to EBCDIC, and causing an error message to be sent to the originating station when the incoming group detects any user-specified error in the segment. In performing its functions, the incoming group of the MH scans and processes header fields based upon the relative order of the individual MH macro instructions (the order in which the incoming group scans the header fields may be varied so that subsequent operations performed in the buffer containing the message segment may be varied). The incoming group also can check the destination code before routing the message to the destination queue.

Step 5: After processing by the incoming group, the message is placed on a destination queue for either an application program for processing, or an accepting station. (If there is to be no processing of the message, proceed with Step 13). The destination queue for an application program may be on a DASD (secondary) or in main (primary) storage, and it consists of queues of message segments that are to be transmitted to a certain application program; see the discussion of the message queues data set in an earlier section, *Data Set Definition and Control Information*. All messages requiring text processing are routed to the destination queue for the application program that processes that type of message. The user controls this routing via the message header (by placing the name of the destination queue for the application program in the destination field in the message header) and by MH macro instructions that direct messages of a particular type to a particular queue.

Steps 6, 7, and 8: The message from a destination queue for an application program is placed in a main-storage buffer (primary); the outgoing group of an MH (the MH is created especially for the application program and is assigned to it by an operand of a macro in the MCP) places it on the read-ahead queue, a special queue that is used by GET or READ macro instructions issued in the application program.

Step 9: Each time an application program issues a GET or READ, TCAM passes message data from the read-ahead queue to a user-specified work area in the application program. Data is moved in the work unit (see *Application Programs*) specified by the user in his input DCB. As the message data is moved to the work area, TCAM removes the header or text prefix from the buffer. After receiving the message data, the application program processes it as required by the application. An application program may generate a response message to be returned to any station (or perhaps to be sent to another application program for further processing). If response messages are to be sent, a data control block must be defined and opened from the application program to govern message transfer before attempting to PUT or WRITE any message. A DCB macro instruction in the MCP defines the destination queue for the application program.

Steps 10 and 11: When a PUT or WRITE macro is issued, TCAM moves the data from the work area into an MCP buffer (primary storage) before it is handled by the incoming group of the MH specified for the application

program. Header or text prefixes are created when data is moved to the buffer, as they are for other incoming messages. As the message data fills the buffer, TCAM inserts chaining addresses and other control information into the prefix field. The response message generated by an application program can be any user-selected length (it is not limited by TCAM). After the buffer is filled, it is handled by the incoming group of the MH that is assigned to the application program by an MCP macro instruction that provides an interface between the MCP and the application program.

Step 12: If further processing of the message data is required, perhaps in another application program, the message is queued for that destination (and Steps 5 through 11 are repeated). If, however, another application program is not specified as a destination in the header, TCAM places the processed message on the destination queue for the accepting station.

Step 13: The destination queue for an accepting station, like the destination queue for an application program, is a part of the message queues data set and it may be defined to be either in main storage (primary) or on a DASD (secondary). TCAM obtains message segments from the destination queue on a first-ended first-out (FEFO) basis within priority groups.

Steps 14 and 15: The message segment is placed in a buffer and the outgoing group of the MH that is specified for the line processes the message. The MH performs such user-selected functions as converting the code of the message to the transmission code for the station (if necessary), inserting the time and date in the header, message logging, and updating message counts. These operations are performed in the buffers that receive the message segments from the destination queue.

Step 16: TCAM transmits the message, minus the header and text prefixes, to the appropriate station.

Maintaining Orderly Message Flow

This section describes how TCAM maintains an orderly message flow between the central computer and remote stations. Among the factors influencing the flow of messages within a TCAM system are message priority and queuing, and transmission priority. *Message priority* refers to the relative order in which messages are sent out over a line. This order depends upon the priorities assigned to individual messages by the user (as specified by a priority field in the message header and an optional priority macro in an inheader subgroup of the MH), and upon the type of queuing (whether by line or by station). *Transmission priority* refers to the relative order in which messages are sent to and received from a station or stations on a line. The transmission priority (send, equal, or receive) for a nonswitched station is specified by a line group DCB macro that is issued for the line group of which the nonswitched station is a part. The transmission priority for a switched station depends upon whether the computer is calling the station or the station is calling the computer.

Message Priority and Queuing

This section describes the order in which messages are sent relative to each other. This order depends upon three variables:

- whether queuing is by line or by station,
- the relative order in which the messages are received at the destination queue, and
- the priorities assigned to the messages.

Outgoing messages may be queued by destination station or by destination line. When outgoing messages are queued by line, one message queue is created for a line, and messages destined for all stations on the line are placed on this queue. (The incoming group of a Message Handler determines the immediate destination of a message.) Messages are taken off the queue and sent to stations on the line on a first-ended first-out (FEFO) basis within priority groups. That is, messages on the queue that have a higher message priority (as specified in the message header or assigned by an MH macro) are sent before messages having a lower priority; when messages have the same priority, the one whose final segment arrived at the queue first will be sent out first, and the others will be sent out in the order in which their final segments arrived at the queue.

Advantages of queuing by line:

- Queuing by line permits transmission of messages by priority on a line basis to stations on a multipoint nonswitched line; that is, all messages of a given priority on the queue are transmitted before any messages of a lower priority, even if the destinations of the higher-priority messages are two different stations on the line.
- Queuing by line requires less storage space than queuing by station. If he queues by line rather than by station, the user saves at least 65 bytes for each station after the first on a line, plus about 28 bytes per station after the first for each priority level specified beyond one.

When outgoing messages are queued by station, one message queue is created for each station on a line. All messages queued for a given station are sent before any messages queued for other stations on the line. (The user can modify this technique for nonswitched lines by using a TCAM macro instruction to limit the number of messages to be sent to a particular station.) Messages on a queue are sent to a station on first-ended first-out (FEFO) basis within priority groups. That is, higher-priority messages are sent before lower-priority messages; when two messages on a queue have equal priority, the one whose final segment arrived at the queue earliest is sent first. For a multipoint line, the relative order in which the *queues* of messages are transmitted is also determined on a FEFO basis; that queue containing the message whose incoming transmission over

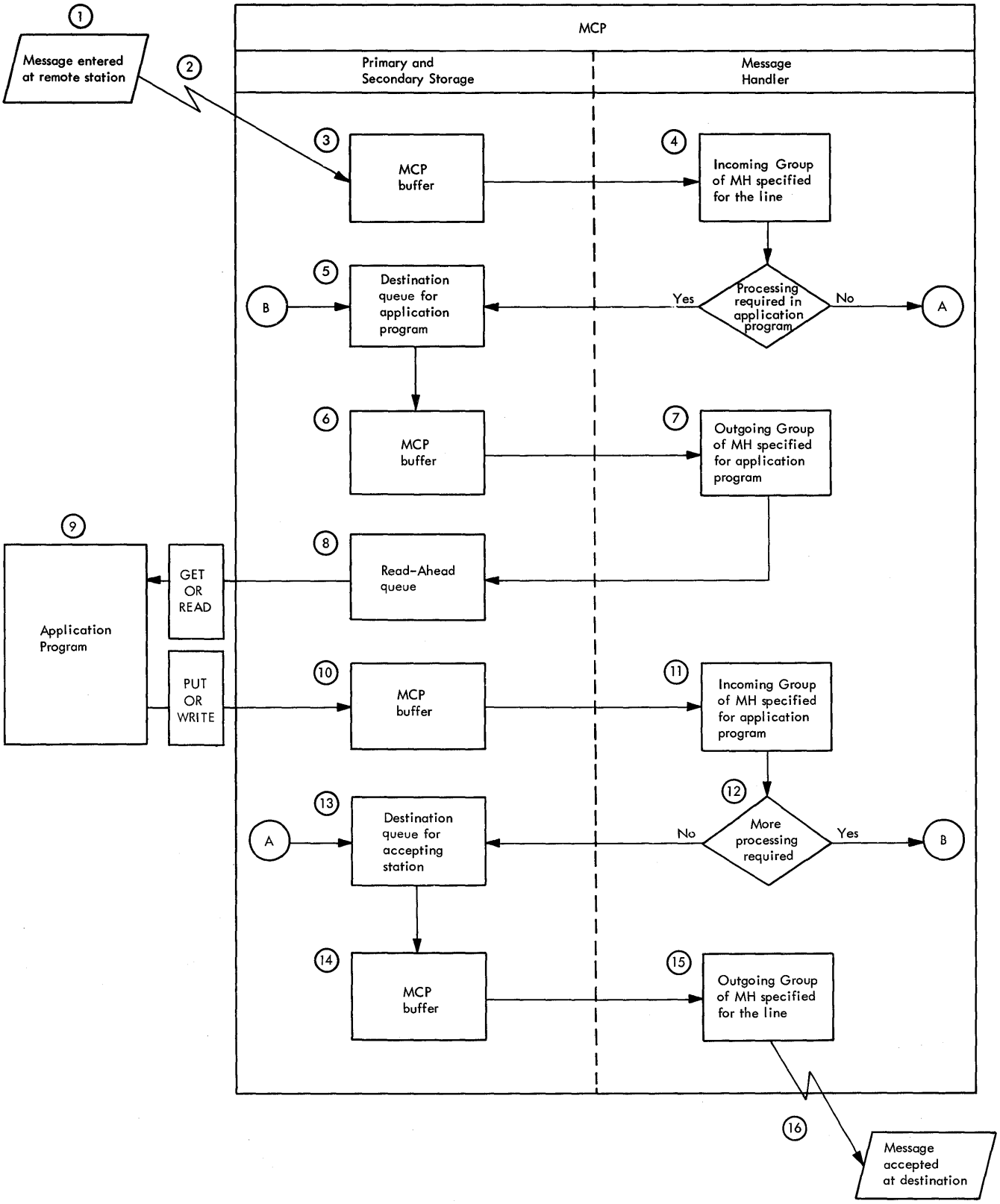

Figure 5. TCAM Message Flow

the line was completed first will be sent before any other queue for a station on that line is sent.

Queuing by station must be specified for switched stations and for buffered terminals.

Messages to be sent to an application program are placed on a queue for that program, and are removed from it as if they were messages queued by station; that is, they are sent to the application program on an FEFO basis within priority groups.

Advantages of queuing by station:

- Queuing by station permits transmission of messages by priority on a station-by-station basis; that is, all messages in a given queue for a station on a line are transmitted before any messages in other queues for stations on the line are transmitted, whether or not the other queues contain messages having priorities higher than those for the messages being transmitted. Thus, messages for the same station are sent as a group. For nonswitched lines, the user can limit the number of messages sent to a particular station by using a TCAM macro instruction. In this way, he can ensure that higher-priority messages to other stations are not excessively delayed.

Transmission Priority

Transmission priority refers to the relative order in which messages are sent to and received from the stations on a line; specification is on a line-group basis by means of the line group DCB macro.

Normally, message traffic can proceed in only one direction at a time over a half-duplex line. Accordingly, the user has the option of specifying in his line group DCB, for each line group made up of nonswitched lines, a relative priority of receiving versus sending. He can specify that:

- CPU receiving has priority over CPU sending, or
- receiving and sending have equal priority, or
- CPU sending has priority over CPU receiving.

Transmission priority has a different meaning for each of three configurations of stations supported by TCAM:

1. polled stations on a nonswitched point-to-point or multipoint line;
2. contention stations on a nonswitched point-to-point line;
3. stations on a switched line.

Nonswitched Polled Stations: For such stations, the user may specify that sending has priority over receiving, that receiving has priority over sending, or that sending and receiving have equal priority.

If receiving has priority over sending, TCAM polls all stations on the line designated in the invitation list as active for entering messages. In polling, TCAM begins with the first active station specified in the list, and proceeds sequentially until a negative response to polling is received from the last active station in the list. Each entry in the list is polled until a negative response is received at the CPU, or until a user-specified limit of messages is reached. At the end of the list, TCAM observes an invitation delay as specified by the line group DCB macro or by an operator command. During the invitation delay, outgoing messages may be sent to stations on the line. (If there are no messages to send at this time, the specified delay is observed nevertheless.) Outgoing messages are sent either until the delay expires or the destination queues for stations on the line are empty. Upon expiration of the delay, outgoing message transmission ends after the current message is sent, regardless of whether any messages still remain queued. As soon as outgoing message transmission ceases, polling and incoming message transmission resume, and the cycle is repeated. It is important to note that if no invitation delay is specified, outgoing message transmission does not occur. If an invitation delay is specified, it must be long enough to accommodate the expected density of outgoing message traffic; too short a delay causes outgoing messages to accumulate on the destination queues for lines or stations in a line group, leading to decreased communication efficiency.

If receiving and sending have equal priority, polling and incoming message traffic proceed without interruption until all active stations on the line have been polled (i.e., until the end of the invitation list is reached). Then outgoing messages, if any are present on the destination queues for the stations on the line, are sent to those stations. Once outgoing transmission begins, it continues until all messages queued for stations on the line have been sent, regardless of whether the user has specified an invitation delay. When all messages for stations on the line have been sent, polling and incoming message traffic resume. Note that, in contrast to the case where receiving has priority over sending, outgoing message transmission occurs whether or not an invitation delay is specified and regardless of the specified length of the delay.

If sending has priority over receiving, outgoing messages (if any are queued for stations on the line) are sent:

1. Each time a negative response to polling is received from a station.
2. Each time an end-of-transmission (EOT) character is received from a station, indicating that a complete message has been sent.
3. Each time the end of the invitation list is reached.

Once outgoing message transmission begins, it continues until all messages queued for stations on the line are sent. Note that when sending has priority over receiving, outgoing transmission can occur after each station is polled, rather than only after a complete polling sequence.

For lines polled under the control of the Auto Poll machine feature, the scheme is slightly modified. If receiving has priority over sending, or if receiving and sending have equal priority, there is no difference between Auto

Poll and programmed polling. If sending has priority over receiving, outgoing messages are sent over lines using the Auto Poll feature:

1. Each time an end-of-transmission character is received from a station, indicating that a complete message has been received at the CPU.
2. Each time the end of the invitation list is reached.

For lines to polled buffered terminals, either equal or send priority should be specified in the line group DCB macro. For buffered terminals, send priority has the meaning described above for other polled stations, except that message segments rather than complete messages are sent (see below). When equal priority is specified, message segments are sent when the end of the invitation list is reached, but only until the specified invitation delay has elapsed and transmission of the current segment has completed, at which time polling is restarted.

Transmission of messages to buffered terminals is handled on a segmental basis: the first segment of a message is sent; if there are messages queued for other stations on the line, their first segments are sent; subsequent segments are sent as stations become available for accepting. For a buffered terminal to become eligible to receive another segment, a time interval (specified by a TCAM macro that causes the buffered terminal to be included as an entry in the terminal table) must have elapsed.

Nonswitched Contention Stations: For nonswitched contention stations, either send or equal priority may be specified. The stations included in this category are the IBM 2740 Communication Terminal (Basic), the IBM 2780 Data Transmission Terminal, the IBM 2770 Data Communication System, and the IBM System/360, System/360 Model 20, and 1130 computers used as terminals.

Send priority is similar for all six types of stations; if send priority is specified, messages may be entered at the station when the line is currently active but not being used. When a message is queued for sending, TCAM checks to see whether a message is being entered by the station; if so, the computer waits until an end-of-transmission control character is received, terminates the receive operation, and then sends all messages queued for the station. If no message is being entered, the computer sends all queued messages immediately after terminating the receive operation. After sending all messages, the computer is again ready to receive messages from the station.

The way in which equal priority works is device-dependent; equal priority is described here for the IBM 2740 Communication Terminal (Basic). If equal priority is specified on a nonswitched contention line, messages may be entered at the terminal when the line is currently active but not being used. If the invitation list for this line consists of one dummy entry, the terminal operator may ask the computer to send by pressing the BID key and then pressing the EOT key. The computer then sends all messages queued for the terminal. (If there is more than one dummy entry in the invitation list, the terminal operator must press BID-EOT once for each entry before the computer will send.) After sending all messages, the computer is again ready to receive messages. Messages queued for the terminal are also sent as soon as the terminal operator enters a number of consecutive messages in the sequence *BID key-message-EOT*, which is equal to the number specified by a macro coded in the inheader subgroup of the Message Handler for this line.

Switched Stations: For switched stations, send priority must be specified in the line group DCB macro.

The relative order in which messages are sent to and received from a station on a switched line depends upon whether the station dialed the computer or the computer dialed the station.

If the station called the computer, the station enters all the messages it has ready before the computer sends any messages. After the station has entered all the messages it has ready, the computer sends all messages queued for the station (only if the calling station has identified itself by means of a unique ID sequence or an origin field in a message header). This cycle is repeated as long as either the computer or the station has one or more messages to transmit. When the last incoming message is received and no further messages appear on the destination queue for the station, the computer breaks the line connection, making the line available for new calls.

When the computer calls the station, the computer sends all messages queued for the station before the station enters any messages (the Auto Call feature must be used when the dialing operation originates at the central computer). Once all queued messages are sent, or if the queue is empty, the station enters all messages it has ready. The cycle of sending and entering is repeated until the station indicates that it has no more messages to enter, and no further messages appear on the destination queue. At this point TCAM breaks the line connection, making the line available for new calls.

Message Handler Functions

Functional macro instructions perform the specific operations that are required for message segments being handled by the various subgroups of an MH. Message segments are directed to the appropriate subgroup by the delimiter macros in the MH; the functional macros within the subgroup are then executed in the order in which they are specified. Functions provided by an MH include:

- message editing (insertion of date, time, and sequence number; insertion or removal of characters or character strings);
- validity checking (verification of source and destination codes and of sequence numbers in incoming message headers);

- routing messages to various destinations or alternate destinations, possibly by priority;
- maintaining counts and logs for terminal message traffic; and
- error checking and handling (checking for errors in transmission and taking corrective action).

General Information

As described previously, a TCAM message may consist of header and text portions. The header portion is the primary concern of the MH sections of the MCP. If any processing of the text portions of messages is required, this processing may be performed by an *application program*, written by the user to suit the needs of his particular application. Application programs run asynchronously with the MCP, usually in another partition or region, but always as a separate system task.

Messages to be processed are routed to a destination queue by an MH; a destination queue and the corresponding process entry in the terminal table are defined by a macro instruction that is coded with the group of macro instructions that define the terminals and line control areas of the MCP. A message from any station (or from an application program) is routed to any predefined application program by a TCAM macro coded in the incoming group of an MH.

TCAM application programs allow the user to define at execution time, via a parameter in the job control DD statement for the application program DCB, which of the destination queues specified in the terminal table is to be linked to the related data set. The user provides DCB macro instructions to define the interface between application programs, OPEN macro instructions to activate that interface, and CLOSE macro instructions to deactivate it; OPEN and CLOSE macro instructions should be issued for each data set specified.

The main concern of TCAM with respect to an application program is to pass messages to the program for processing and later to return the messages to the appropriate destination. (However, there may be no return message, as in the case of a file update application.) TCAM provides the means of transferring data between the partitions (GET, PUT, READ, WRITE, and CHECK macros), and provides a special scheme for buffer usage for application programs.

In some applications, the required processing may be such that one destination queue can handle all the messages, and a single application program can perform the processing. If various kinds of processing are required, there are two ways of providing it:

- A destination field in the message header may be used to route the message to the destination queue corresponding to the appropriate program.
- Alternatively, all messages that require processing can be routed to the same application program, where a user-written analysis routine can determine the kind of message received, based upon a user-specified code in the message; the messages can be transferred by this routine to the appropriate processing routines, or possibly to a processing program in another partition or region (via a PUT or WRITE to a destination queue).

When the destination field in the header is used to route messages to the appropriate processing program, MH facilities are used to determine the MH processing needed for the message. Messages requiring different handling in the MH can be differentiated by means of two TCAM-provided macro instructions.

The GET or READ macros that obtain messages from the destination queues transfer the data to a user-specified work area. The work area and the units of work placed in it are discussed below. Once in the application program work area, the data can be analyzed and processed by the application program. Optionally, a PUT or WRITE macro causes a response message to be returned to the MCP for transmission either to a station (not necessarily the one that originated the message), to a list of stations, or to another application program.

Two kinds of application programs may be run with a TCAM MCP: those written to run with the Queued Telecommunications Access Method (QTAM), and those written to run under TCAM that are compatible with the Sequential Access Method (either BSAM or QSAM).

Data Independence

QTAM and TCAM application programs need not be concerned with the station at which a message originated, or with the transmission code of the line, or with what the station line control had been. TCAM provides the facility for inserting and removing line control in the MCP.

Message Flow to an Application Program

This section describes the flow of a single-segment message between a station and an application program operating under TCAM. The steps described here are repeated for a multisegment message, except that the response message, if any, may be returned by a PUT macro any time after the first segment is received. This discussion summarizes the description in the section titled *Message Flow within the System*, and includes a detail unique to application programs, the read-ahead queue.

A message enters the MCP and is placed in a buffer. The buffer is handled by the incoming group of the MH specified for the line and is placed on the destination queue for the application program. (See Figure 5, steps 1-5.)

The buffer then is handled by the outgoing group of the MH specified for the application program. At this point, the message is queued on the read-ahead queue, an area in main storage from which the application program obtains the message by issuing GET or READ macro instructions. These macros obtain the messages in units of data that fit an area of the application program called the *work area*. The message is placed in the work area for processing. The

size of the work area bears no relationship to the size of the MCP buffers. (See Figure 5, steps 6-9.)

After processing, and assuming there is a response message, the message is returned to the MCP, where it is placed in buffers. The buffers are handled by the incoming group of the MH specified for the application program and are placed on the appropriate destination queue for an accepting station (assuming there is no more processing to be done on the message). After handling by the outgoing group of the MH for the line, the response message is sent to its destination. (See Figure 5, steps 10-16.)

QTAM Application Programs

Application programs written to run with a QTAM MCP may be adapted to run with a TCAM MCP. This section does not describe the QTAM services available (see the publication *IBM System/360 Operating System Queued Telecommunications Access Method Message Processing Program Services*, Order No. GC30-2003). Rather, the section describes the manner in which QTAM application programs must be modified to run under TCAM.

QTAM application programs being modified to run under TCAM need only be reassembled with a QSTART macro as the first instruction. During execution, the modified application program operates in most respects as it did under QTAM. GET and PUT macro instructions cause data transfer between the partitions. There is a GET/PUT prefix and a work area. Message, record, and segment logical units are handled. RELEASEM. CLOSEMC, CKREQ, STARTLN, STOPLN and RETRIEVE macro instructions maintain their functions as in QTAM. Other macro instructions (COPYQ, COPYP, COPYT, CHNGP, and CHNGT) are each assembled as a NOP (however, their functions are available through either the operator control facility, described in the next chapter, or through TCAM macros). The checkpoint facility is supported.

Thus, an installation can continue to use QTAM application programs when conversion is made from QTAM to TCAM. (Any new functions or application programs will use the TCAM application program facilities.)

Checkpointing the QTAM Application Program

The TCAM checkpoint facility for QTAM application programs running with a TCAM MCP performs a different function from that performed when the application program runs under QTAM. The application program issuing the CKREQ macro formerly placed the application program partition in the wait state until a certain number of CKREQ macros were issued in different application programs. It now waits only long enough for the destination queues for that program to be checkpointed. See the *TCAM Checkpoint/Restart Facility* discussion in the next chapter for a more detailed description of the services provided by the TCAM checkpoint/restart facility.

Reassembling the QTAM Application Program

To run a QTAM application program under a TCAM MCP, the user must add a QSTART macro instruction and reassemble the program.

Those macros which are assembled as a NOP may, if desired, be replaced by the corresponding TCAM macro. However, due to the differences between the formats of QTAM and TCAM control areas, user code that handles the control areas will also need to be modified. As an alternative to using TCAM macros, operator commands may be used to replace the QTAM macro functions. Corresponding macros and operator commands are shown in Figure 6.

QTAM MACRO	TCAM MACRO	OPERATOR COMMAND	NOTE ON RESTRICTED FACILITIES OFFERED BY OPERATOR COMMANDS
COPYP	ICOPY	ACTVATED INACTVTD STATDISP	Lists active and inactive stations and displays invitation list, respectively
COPYQ	QCOPY	QSTATUS RLNSTATN	Displays only the relative line number of a station
COPYT	TCOPY	OPTFIELD STSTATUS	Displays the option fields Displays the status byte and the sequence number
CHNGP	ICHNG	AUTOSTOP AUTOSTRT ENTERING NOENTRNG	Only change individual fields
CHNGT	TCHNG	DATOPFLD	Only changes individual option fields

Figure 6. Corresponding Macro Instructions and Operator Commands

TCAM Application Programs

The introductory section, *General Information*, is applicable to TCAM as well as QTAM application programs. TCAM provides certain network control facilities from an application program as well as a means of message retrieval from the MCP by means of the Operating System POINT macro with GET or READ macro instructions. All of the functions of the operator control facility are available from application programs; operator commands may be PUT to the MCP.

Data is transferred between the MCP and the application programs using either BSAM (READ, WRITE, and CHECK macro instructions) or QSAM (GET and PUT macro instructions). When using QSAM, either *move* or *locate* mode may be specified in the input and output DCB macro instructions. BSAM implies move mode. For a discussion of the move and locate processing modes, see *IBM System/360 Operating System Supervisor and Data Management Macro Instructions*, Order No. GC28-6647, and the glossary in this publication.

The SYNAD exit for logical errors is handled in the same manner as it is handled under BSAM and QSAM. The SYNADAF and SYNADRLS macros may be used. POINT is used only in conjunction with GET or READ macros for message retrieval.

Certain other features can also be incorporated into an application program:

- A PUT or WRITE work area prefix can be used to specify the destination to which a message can be sent.
- A GET or READ work area prefix can be used to receive the name of the message source.
- The work area contents may be described to TCAM for PUT or WRITE operations and by TCAM for GET and READ operations as first-segment, intermediate-segment, last-segment, or single-segment message.

These options may be included at execution time by a job control DD statement parameter.

Work Area

When moving records in the move mode, the user must define in the application program an area to be used for handling the work units obtained by GET or READ macro instructions. The size of the work area must be specified in the input DCB. If locate mode is specified in the input DCB macro instruction, the size of the work area is *not* specified and TCAM provides the work area, allocating from the application program's main storage a work area equal to the number of bytes specified in a keyword operand of the input DCB.

The work area should be as long as the record length specified in the input DCB. Optionally, an additional number of bytes may be specified for the identification of the message source (input DCB) or destination (output DCB), an indication of the work unit type (supplied by TCAM), and specification of variable length records (4 bytes or 8 bytes for unblocked or blocked records, respectively). The maximum size of the work area is 65,535 bytes.

Work Unit

The work units handled by a TCAM application program are either messages (fixed format only) or records (see note below). A *message* is a unit of text that ends with an end-of-transmission character. Because line control characters are deleted by the MCP, the buffer prefix contains an indication of the end of the message. A *record* is a unit of text being moved into the work area; the work area terminates when TCAM encounters one of the following conditions:

- the work area is filled, or
- the length indicated in an operand of the TCAM input DCB macro is reached (fixed-length records only), or
- a user-specified delimiting character is found. This character is specified by a TCAM editing macro in the MCP.

Note: Records may be either fixed-length, variable-length, or undefined-length. The length of fixed-length records is determined by a value specified in the input DCB. Variable-length and undefined-length records are determined by the user-specified delimiter, by the work area being filled, or by the end of the message.

Message Control Program Interface

The TCAM Message Control Program (MCP) routes messages between an application program and stations. Because an application program depends on the MCP to perform its I/O operations, an interface must be established between an application program and the MCP. TCAM provides the following services that allow this interface to be established from an application program:

- definition of the interface (via the input and output DCB macro instructions);
- initialization and activation of the interface (via the OPEN macro instruction);
- transfer of messages between the application program and the MCP (via GET, PUT, READ, WRITE, CHECK, and POINT macro instructions); and
- deactivation of the interface (via the MCPCLOSE and the CLOSE macro instructions).

TCAM also provides buffer facilities specifically designed for the MCP interface. Unlike the functions performed by the analysis and processing routines of an application program, these functions are partially or wholly peculiar to TCAM and the teleprocessing environment. Therefore, TCAM provides routines to accomplish these functions. Linkage to these routines is established by TCAM macro instructions in an application program.

Sequential Access Method (SAM) Interface

Application programs accomplish data transfer to and from the MCP by using either GET/PUT (QSAM) or READ/WRITE/CHECK (BSAM) macro instructions. Record formats may be fixed, variable, or undefined.

The user can write an application program that runs under TCAM just as he would a QSAM or a BSAM program. Optionally, the application program can include other features. The destination of the message can be specified in a PUT (or a WRITE) work area prefix. The application program can also receive the name of the

message source in a GET (or a READ) work area prefix. The application program may also request, or indicate to TCAM, the description of the work area contents in TCAM terms; that is, first-segment, intermediate-segment, last-segment, or single-segment message. These options may be included at execution time by a job control DD statement.

TCAM supports a telecommunication system with some facilities that are specified by the user; other facilities are provided automatically by TCAM.

Operator Control

The operator control facility provides a wide variety of functions that enable the user to examine or alter the status of the telecommunication network during execution. Control facilities are available for an application program and the system console as well as for a remote station; these facilities are supported by either resident or nonresident routines.

The functions are:

- activating and deactivating stations;
- activating and deactivating lines;
- switching between Auto Poll and programmed polling;
- activating and deactivating invitation list entries;
- starting and stopping the TCAM I/O interrupt trace facility;
- releasing messages queued for a station;
- holding messages in the queue for a station;
- interrogating the status of a line, queue, or station;
- initiating the system delays;
- altering the duration of either the programmed polling delay or the system delay;
- switching the primary operator control station;
- altering option fields;
- debugging the TCAM environment; and
- initiating system closedown.

The user may request these functions by entering an operator command either from a specific nonswitched station that has been designated an operator control station, or from either the system console or an application program. Such a message is identified to TCAM by a special identification sequence when it is entered from either a station or an application program (an operator command entered from the system console does not require this identification sequence). Facilities are provided for changing the primary to one of any number of secondary control stations, for returning incorrectly formatted commands, and for providing responses in English (unless the operator command is rejected because of cancelation or incorrect formatting).

Note: Operator commands from stations and application programs must be contained in a single line buffer; if the buffer is too small, the command is truncated and an attempt is made to process it. Commands from the system console must be no longer than 126 characters.

Operator Control Stations

Various nonswitched stations, including the system console, may be designated operator control stations. Possible operator control terminals are the IBM 1050 Data Communication System (nonswitched), IBM 2740 Communication Terminal (Basic), IBM 2740 Communication Terminal (with Station Control), AT&T 83B3 Selective Calling Stations, Western Union Plan 115A Outstations, and IBM 2780 Data Transmission Terminal (nonswitched). The system console is defined at system generation time and may include a card reader and other I/O devices. An *operator control station* can generate operator commands and receive related response messages. The user may define one primary control station and any number of secondary control stations.

A *primary* control station can receive internally-generated error recovery procedure (ERP) messages indicating that a permanent I/O error has occurred; it also has the capabilities of a secondary control station. A *secondary* control station can send control messages and receive related responses, but not ERP messages (with one exception: when a primary operator control station other than the system console becomes inoperative, an internally-generated error message is sent to the system console, a secondary operator control station in this situation, stating that the primary operator control station is inoperative). TCAM allows primary control to be transferred to one of the secondary control stations. The system console may be specified either as a primary or as a secondary station; if neither is specified, it is primary by default. The station operator may cancel a partly entered control message. When specifying either a station or a line, care must be taken not to perform an operation on a line that includes the primary station if the operation would stop input or output to the station.

Operator Commands from an Application Program

An application program may enter an operator command by moving the command into its PUT or WRITE work area (it has the same format as operator commands that are entered from stations). A PUT or WRITE is issued to move the messages from the application program to the MCP, where each message is directed to the subgroup in the incoming group of the MH that handles messages entered by the application program.

Once it is recognized in the subgroup, the operator command from the application program is treated like any other operator command.

The response to an operator command entered by an application program is directed to any specified station that is defined as an operator control station.

TCAM I/O Error-Recording Facility

TCAM provides an I/O-error recording facility that creates records on disk when certain terminal-related I/O errors occur. The TCAM error-recording facility, an extension of the OS/360 outboard recorder (OBR) and statistical data recorder (SDR) error-recording programs, helps to reduce the time that the TCAM system is inoperative by providing information useful in diagnosing line and station problems.

TCAM maintains two internal counters for certain stations and lines. The *stations* that are associated with these internal counters are all the switched and nonswitched stations that can accept messages (and some terminals that can only enter messages); also included is each group of nonswitched terminals equipped with the hardware group-code feature (terminals can only accept messages under the group-code feature—they cannot enter messages). The *lines* that are associated with these internal counters are all switched lines to stations that do not uniquely identify themselves after calling the computer. One of the two counters is a two-byte counter that keeps track of the number of Start I/O (SIO) commands issued by the CPU for the station or line. The other is a one-byte counter that contains the number of temporary errors (defined errors occurring during SIO operations for which retry was successful) that have occurred since the last error record was written on disk. If the station for which an SIO operation is being performed is known, the counters for that station are updated. The counters for a line are updated only if the station for which the SIO operation is being performed is not known; the counters are reset by the system each time their contents are recorded on disk.

Four types of I/O error records may be written on disk; permanent, temporary, overflow, and end-of-day.

- A *permanent-error record* is written on disk for each permanent I/O error. A permanent I/O error is either an undefined, unanticipated I/O error for which TCAM provides no error recovery procedure, or an I/O error for which TCAM provides an error recovery procedure that has tried several times to correct the error and failed on each trial.
- A *temporary-error record* is made on disk when an error specified for a particular line or station by an operator command occurs, provided that TCAM's error recovery procedures are successful in causing recovery from the error. If TCAM's error recovery procedures are unsuccessful, a permanent-error record is made, and a special message is sent to the primary operator control station; see *Operator Awareness Message* following. This record contains the same information as the permanent-error record.
- A *counter-overflow record* is made when either the SIO counter or the temporary-error counter is about to overflow. Each record contains the same information as the counter overflow record.
- When a line group data set is closed, an *end-of-day record* is made for stations and lines in the line group.

Intensive-Mode Error Recording

A station or a line in *intensive mode* is one for which a temporary-error record is made each time that a specified error occurs and is recovered from. Either a station or a line may be put in intensive mode with an operator command. In issuing the operator command, the user may specify one of ten types of errors checked for by TCAM and the number of times (from 1 to 15) that a temporary-error record is to be made when the specified error occurs and is recovered from for this line or station. Alternatively, the user may specify that a temporary-error record is to be made if *any* of the ten types of errors occurs (except for an unusual leading graphics response from the IBM 2740 Communication Terminal, Model 2) and is recovered from; in this case he would also specify the number of times a recording is to be made.

Intensive-mode error recording may be specified either for a line or for a station. If a station is specified, temporary-error records are made when the error specified in the operator command occurs for that station and is recovered from. If a line is specified, a temporary-error record is made each time the specified error occurs and is recovered from for any station on the line (that is, all stations on the line are placed in intensive mode).

If a station is placed in intensive mode for one type of error, and the operator command is issued a second time for the station, but this time specifying a different type of error, the type of error specified by the second operator command is the one that causes temporary-error records to be made after it is issued. An operator command for a line overrides any that have been issued previously for stations on the line. For example, if an operator command specifying that temporary-error records be taken for data checks occurring for a station named NYC is followed by an operator command specifying that temporary-error records be taken for time-outs occurring for any station on the line, after the second operator command is issued, temporary-error records are no longer taken for data checks occurring for the station NYC. If a third operator command specifying that temporary-error records be taken for data checks occurring for NYC is now issued, a temporary-error record is made each time a data check occurs for NYC, and each time a time-out condition occurs for any other station on the line.

Operator Awareness Message

A message is sent to the primary operator control station if an I/O error occurs for which TCAM provides error recovery procedures, but the error recovery procedures are unsuccessful (this is in addition to the permanent-error record that is created on disk when such an error occurs).

Gaining Access to Error Records

Permanent, temporary, counter-overflow, and end-of-day TCAM I/O error records are located in a data set that the user can gain access to by using the IFCEREPO system utility program. Information on using the IFCEREPO program to write TCAM I/O error records is found in the description of this utility in the publication *IBM System/360 Operating System Utilities*, Order No. GC28-6586.

Debugging Aids

TCAM provides, in addition to the standard Operating System dumps of the MCP partition or region, some special aids for debugging the telecommunications network and the MCP. These aids include:

- The *TCAM formatted ABEND dump*, which is provided automatically when the TCAM MCP fails (this TCAM dump is in addition to the system ABEND dump that is provided). It formats TCAM control areas and subtask control areas that are attached to the TCAM MCP partition or region.
- The *I/O error-recording facility*, which creates records on disk when certain terminal-related I/O errors occur. This facility is an extension of the OS/360 outboard recorder (OBR) and statistical data recorder (SDR) programs and helps to reduce the time that the TCAM system is inoperative by providing information useful in diagnosing line and terminal problems.
- The *TCAM logging facility*, which enables the user to maintain a record of incoming or outgoing message traffic on a sequential medium (tape or disk).

Other optional debugging aids include building a *cross-reference table* in main storage, which provides the user with a convenient means of locating in a standard Operating System dump certain information associated with each open line, and obtaining special dumps of:

- The *TCAM I/O interrupt trace table*, which provides a sequential record in main storage of the I/O interrupts occuring on a specified line.
- The *dispatcher subtask trace table*, which is used to keep a sequential record in main storage of the subtasks activated by the TCAM dispatcher. An entry is placed in the table by the TCAM dispatcher each time a TCAM subtask is dispatched.
- The *buffers*: the buffer dump consists of writing buffer and status information to either magnetic tape or disk (there is no table in main storage that contains buffer and status information).
- The *message queues data set*: a dump of the message queues data set may be dynamically invoked via a separate TCAM utility. This utility formats the DASD message queues data set for immediate printing, or directs this data set to either magnetic tape or disk for later printing.

In addition, the cross-reference table, the STCB trace table, and the I/O interrupt trace table reside in main storage and may be included in a standard Operating System dump.

Alternate Destination Definition

The TCAM user may issue a functional macro in either the incoming group or the outgoing group of an MH to indicate an alternate path to a station as well as to indicate an alternate destination station; any single, group, or cascade list entry in the terminal table that defines an alternate path may be specified. This optional macro is executed when specified error conditions are detected.

Network Reconfiguration

Operator commands and application program macro instructions perform the various functions required to support network reconfiguration for teleprocessing devices. Network reconfiguration facilities can start or stop lines or line groups, start or stop a station on a line, remove a station from an invitation list, cause transmission to resume for an intercepted station, or hold up transmission to a station. These functions provide sufficient flexibility for any user who wishes to change, for example, the status of a defective station before performing a manual device reconfiguration.

TCAM Checkpoint/Restart Facility

The TCAM Checkpoint/Restart facility allows the TCAM MCP to be restarted following closedown or system failure by periodically recording in a special data set on disk certain information on the status of each station, destination queue, terminal table entry, and invitation list in the system. When start-up after system closedown or failure occurs, TCAM uses this information to restore the MCP environment to its condition before closedown or failure. Upon restart, the terminal table, option table, invitation lists, and internal control blocks associated with stations and lines are restored to the condition they were in just before the last checkpoint record was taken; outgoing message traffic to each destination resumes with the highest-priority unsent message.

The TCAM checkpoint/restart facility permits restoration of the MCP environment upon restart. The OS/360 advanced checkpoint/restart facility (described in the publication *IBM System/360 Operating System Advanced Checkpoint/Restart Planning Guide*, Order No. GC28-6708) may be used to perform a similar service for TCAM application programs. TCAM design permits checkpoints of the MCP to be coordinated with Operating System checkpoints of TCAM application programs, so that the entire TCAM system (MCP plus application programs) can be restored as nearly as possible to its condition at the time of closedown or failure. These features are discussed in the chapter *Writing TCAM-Compatible Application Programs* in the *IBM System/360 Operating System TCAM*

Programmer's Guide and Reference Manual, Order No. GC30-2024.

How the TCAM Checkpoint Facility Works

Checkpoint records containing the information necessary to reconstruct the MCP environment upon restart are kept in the checkpoint data set on a DASD. The four types of records that may reside in the checkpoint data set are a control record, two or more environment records, a series of incident records, and one or more checkpoint request records:

- The *control record* is used internally by TCAM during restart and requires no user programming considerations.

- *Environment records* are used to record the total MCP environment; each environment record contains information on the status of each message queue, line, terminal, and (optionally) invitation list at the time the record was taken, and also includes the contents of the option fields for each station.

  Environment records are taken automatically at certain points during the execution of the MCP and ensure satisfactory restart of the MCP after system closedown or failure. If the user is synchronizing his TCAM checkpoints with OS checkpoints of the application program, he may wish to ensure that a TCAM environment checkpoint be taken at some time interval after the last environment checkpoint. The user may do this by specifying the time interval in a macro instruction also specifies the number of environment records he desires to keep in his checkpoint data set at any one time. He may request, for instance, that the three most recent environment records be kept in the checkpoint data set. When a new checkpoint record is taken, it overlays the oldest environment record in the data set. Ordinarily, TCAM uses the most recent environment record in the data set to reconstruct the MCP environment for a restart. If, however, the latest record cannot be used (due, perhaps, to a disk read or write error), TCAM informs the user with a WTO message at the system console and automatically attempts to use the next most recent record. If that record is also unusable, and if there is another environment record in the data set, TCAM attempts to use that record.

- *Incident records* are used to record single changes in line status, terminal status, and option fields; these changes occur as a result of execution of MH macros, certain TCAM-related application-program macros, and operator commands. One incident record is automatically taken for each change in line status (from active to idle or vice versa). Each change in station status (from active to inactive or vice versa) is recorded on an incident record when the appropriate keyword operand is coded on the macro instruction that provides most of the TCAM initialization. Each change in a station's option fields caused by either an application program macro or an operator command is automatically recorded on an incident record. One incident record is made of the contents of the option fields assigned to the origin, destination station, or application program each time a message is processed by an incoming or outgoing MH group containing the appropriate macro; this record reflects changes in the station's option fields caused by MH processing of the message. Incident records are used to update environment records at restart time unless the user specifies that a warm restart be performed following either a quick or a flush closedown, or that a continuation restart be performed following system failure (see *Restart* following). The TCAM restart routine takes the latest usable environment record and updates its fields with the contents of all incident records taken since the environment record was taken. In general, an incident record updates only one field in an environment record.

- *Checkpoint request records* are taken when a specific macro instruction (designed to initiate the checkpoint request) is issued one or more times in an application program; they record the status of the application program's message queues, and are used in much the same way as incident records to update the environment record during restart. The number of checkpoint request records set up by the TCAM checkpoint request macros help synchronize TCAM checkpoints with the Operating System checkpoints of the TCAM application programs.

Scanning the Message Queues

In addition to updating the latest usable environment record (or the record specified during TCAM initialization) with any incident records taken since the environment record, the TCAM start-up routine may perform a scan of the message queues in the message queues data set.

A scan of the message queues involves searching the queues from the point at which the environment record being used for restart was taken to the point of system failure. Already-sent messages are passed over, so that after restart occurs, sending of messages to each destination station or application program represented by a message queue resumes with the highest-priority unsent message that was completely received before system failure. Partially-received messages are eliminated.

Closedown

System closedown may be initiated from either an application program or from an operator control station; the two types of closedown that may be initiated from either of these two sources are a flush closedown or a quick closedown.

A *flush closedown* causes any incoming message traffic on each line to cease (the current message being received

on any line is completed). Once the incoming message traffic for each line ceases, any outgoing messages for stations on that line are sent (that is, all unsent messages are *flushed* from the message queues), and an environment record is taken.

A *quick closedown* stops message traffic (incoming *and* outgoing messages) as soon as any current incoming and outgoing message on each line is completed. Queues of outgoing messages are not flushed as they were in the previous discussion; instead, the message status of each queue is preserved with an environment record, and messages are sent to their appropriate destinations after restart.

Restart

Restart is any TCAM start-up other than the initial start-up, and it may or may not involve reconstructing the MCP environment as it existed prior to termination. The type of restart desired must be specified during TCAM initialization.

A *cold restart* ignores the previous environment. The system is started as though for the first time (initial start-up). The message queues are regarded as new and must be reformatted (necessary to detect logical read errors). A cold restart may be used following either a flush close, a quick close, or a system failure.

There are two forms of restart which reconstruct the environment as it existed prior to termination: continuation restart and warm restart. In a *continuation restart*, the MCP environment is reestablished through the use of the last environment record, the incident records, and the checkpoint request records. From this point, the message queues are searched to determine the last complete message received and transmitted for each queue. (If synchronization with Operating System checkpoints of an application program is specified for a process queue, that queue is not scanned during restart for elimination of unsent messages.)

In the *warm restart* following either a quick or a flush closedown, the MCP environment is reconstructed as for a continuation restart. When used following either a quick or a flush closedown, there are no incident records and the search of the message queues is minimal because an environment checkpoint was taken after message traffic was stopped.

The following table summarizes the combinations of termination and restart.

Type of Termination	*Restart Used:*
flush closedown	cold restart
flush closedown	warm restart (Note)
quick closedown	cold restart
quick closedown	warm restart
system failure	continuation restart with queue scan
system failure	continuation restart with *no* queue scan
system failure	cold restart

Note: This may be desired to avoid the loss of messages that could not be flushed during closedown due to a station being inoperative.

Restart is achieved through the same procedure that is used in initial start-up. Continuation following a system failure is accomplished internally unless overridden during TCAM initialization. If either warm restart or cold restart is desired following a quick or flush close, it must be specified during TCAM initialization.

Note: Following either a system failure or a closedown, the MCP must *not* be reassembled prior to continuation and warm restarts. If the MCP is to be reassembled, a cold restart must be used.

On-Line Test Function

The on-line test function is an optional TCAM facility that permits either a system console operator or a remote station user to determine whether transmission control units and remote stations are working properly by:

- diagnosing machine errors,
- verifying repairs,
- verifying engineering changes,
- checking devices periodically, and
- checking new stations brought on-line.

No programming considerations are required of the system console operator or the remote station user who requests an on-line test for a device. However, the system programmer must specify during TCAM initialization that this optional TCAM function either is or is not wanted. An algorithm is provided to determine the amount of main storage required for the on-line test control program (this control program also acts as an interface between TCAM and the device tests, conveys messages to the user about the tests, schedules and controls tests, and prompts the user either when he requests help or makes an invalid request, or when a test needs more data). The minimum amount of main storage for running an on-line test is 10K bytes.

Detailed user information is contained in the *TOTE/Configurator User's Manual*, part 5172900, which can be obtained through your local IBM branch office.

A teleprocessing system operating under TCAM can be designed for a wide variety of applications as part of message control and/or message processing within the application program.

Message Control

Three telecommunications applications suited to handling by an MCP are those of data collection, inquiry handling, and message switching.

Data Collection

Data collection can be accomplished entirely within the MCP. Stations enter data to the computer in the form of messages. The messages are accumulated and stored by the computer, and subsequently processed as a batch.

The MCP can accumulate data in two ways:

1. By storing the data on destination queues for application programs.
2. By storing the data on any secondary storage medium.

If the first method is used, the messages can be obtained at any time by an application program (see *Processing Collected Data* following). The messages are routed to destination queues in the same manner as any other messages that have an application program as their destination. The messages remain on these destination queues until an application program is activated that issues a series of GET or READ macro instructions to obtain the messages for processing.

In the second method, a logging macro instruction in an MH section of the MCP causes the messages to be recorded sequentially on a secondary storage device selected by the user. An access method other than TCAM must be used to retrieve the messages for processing.

Inquiry Handling

Inquiries are handled by both the MCP and the application program. The MCP receives the inquiry message from the calling station, stores it in buffers, and passes it to the appropriate MH, which issues a macro instruction that allows the station to complete the inquiry without interruption; it then routes the message to the appropriate application program.

The application program returns a response message to the MCP; the MCP routes the message back to the calling station via the appropriate transmission control unit. (User logic in the application program may require that certain inquiry messages not be provided a response; for example, the requested information may be temporarily unavailable, and the application program should provide an appropriate message to the MCP to let the calling station know when it can get the desired information.)

Message Switching

Message switching is accomplished entirely within the MCP. In a message switching application, stations transmit messages to the CPU, which relays the messages to one or more other stations. A message switching application does not prevent a station from sending a message to an application program to be processed. TCAM places these messages on a destination queue for handling by an application program (either concurrently with the message switching application or later).

When an incoming message is to be switched, the MH section of the MCP routes the message to the destination queue for the station to which the message is to be forwarded. If desired, the MH can place information such as the time and date of receipt in the message header. Verifying the validity of the codes for the originating and destination stations, and checking the input sequence number in the header can also be performed. Before a message is transmitted to its destination, the MH can record in the header the date and time the message is sent, and the number of the message relative to other messages sent to that station. The MH also can log the messages on a storage device for subsequent reference by the user with a different access method.

Message Processing

A wide variety of teleprocessing applications can be processed by an application program. Two of these applications are:

1. Processing collected data.
2. Inquiry processing.

Processing Collected Data

Processing collected data is the second part of a two-step application; the first step is the actual collection of the data by an MCP (see a preceding section, *Data Collection.*

If messages are collected on a destination queue, they remain on the queue until an application program issues GET or READ macro instructions to obtain them for processing. The application program that processes the collected data can either:

- operate concurrently with the collection of the data by the MCP; or
- be loaded and initiated at a later time (for example, to process data at the end of the day after all message traffic has ceased).

In the latter case, if the user wishes to have TCAM retrieve the messages from the destination queue, the MCP must remain operational. If the data is collected on a user-selected secondary storage device by a logging macro instruction, the data must be obtained for processing by an access method other than TCAM.

Inquiry Processing

An inquiry application involves receiving messages from stations (performed by the MCP), processing the data contained in the messages (performed by the application program), and sending replies to the originating stations (MCP).

The routine called by the application program to process the messages need not reside in main storage. For example, in an inquiry processing application that requires processing of many different types of inquiries, it may not be economical to have all of the required processing routines in main storage. The application program can contain an analysis routine that determines the type of message and causes the routine required to process it to be loaded from a library.

An optional feature of the inquiry application is operation in "lock mode." In this mode, a station transmitting a message into the system is held on the line by the MCP until the application program generates a response message for transmission back to the inquiring station. The response message is transmitted immediately. Another optional function particularly suited for a high-volume inquiry application is obtained by specifying that the message queues data set be located in main storage rather than on a DASD.

Lock mode is suggested for audio applications. An application program, if used, receives audio inquiries via the MCP Message Handler and processes them according to the requirements of their applications. Once a message has been processed, the application program returns an audio response message to the MCP for forwarding to the calling station.

Multiple Applications

A nonswitched network is used for a multipoint configuration of stations. BSC stations on a multipoint line can be used for different applications; for instance, one station may be used for batch data and another for inquiry applications. Some terminal types may be used for both.

Stations in a multipoint configuration must be mutually compatible:

- polling sequences for all stations must be the same length;
- the invitation sequence for each station must be the same length;
- reaction to line control must be the same for each station; and
- all stations in a multipoint configuration must have the transparent mode facility if any station uses transparent mode.

Appendix A: Machine and Device Requirements

TCAM operates under the IBM System/360 Operating System MFT-II or MVT on any System/360 Model 40 or above (that is, a CPU having at least 128K bytes of main storage). The only additions to the minimum requirements of the System/360 Operating System are:

- All telecommunications terminals, except the IBM 2260-2848 Display Complex (Local), must be attached to either an IBM 2701 Data Adapter Unit Model 1, an IBM 2702 Transmission Control Model 1, an IBM 2703 Transmission Control Model 1, or an IBM 7770 Model 3 Audio Response Unit; they cannot be attached directly to a channel.
- All IBM 2701, 2702, 2703, or 7770 control units that operate under TCAM must be attached to the System/360 via the multiplexer channel. A switch on the CE panel on the 2702 can be used to place a given line in CE mode for equipment checking. Care must be taken to ensure that no lines are in CE mode when TCAM is used, since no ending status will be returned to a Start I/O (SIO) command that is issued by the system.
- No device may be operated in burst mode on the multiplexer channel concurrently with the operation of TCAM, except when the TCAM operation involves only the IBM 2260-2848 Display Complex (Local).

The following additional features may be required:

- ATTACH must be specified for an MFT-II system.
- The line correction feature must be included on IBM 1050 Data Communication System terminals if automatic retry is desired when a transmission error occurs.

TCAM supports any combination of the IBM 7770 Model 3 Audio Response Unit and the IBM 2701, 2702, or 2703 transmission control units on the same multiplexer channel. Up to eight control units can be attached directly to the multiplexer channel. TCAM also supports the IBM 2848 Display Control attached directly either to the multiplexer or a selector channel.

Figure 7 illustrates the device configurations supported by TCAM.

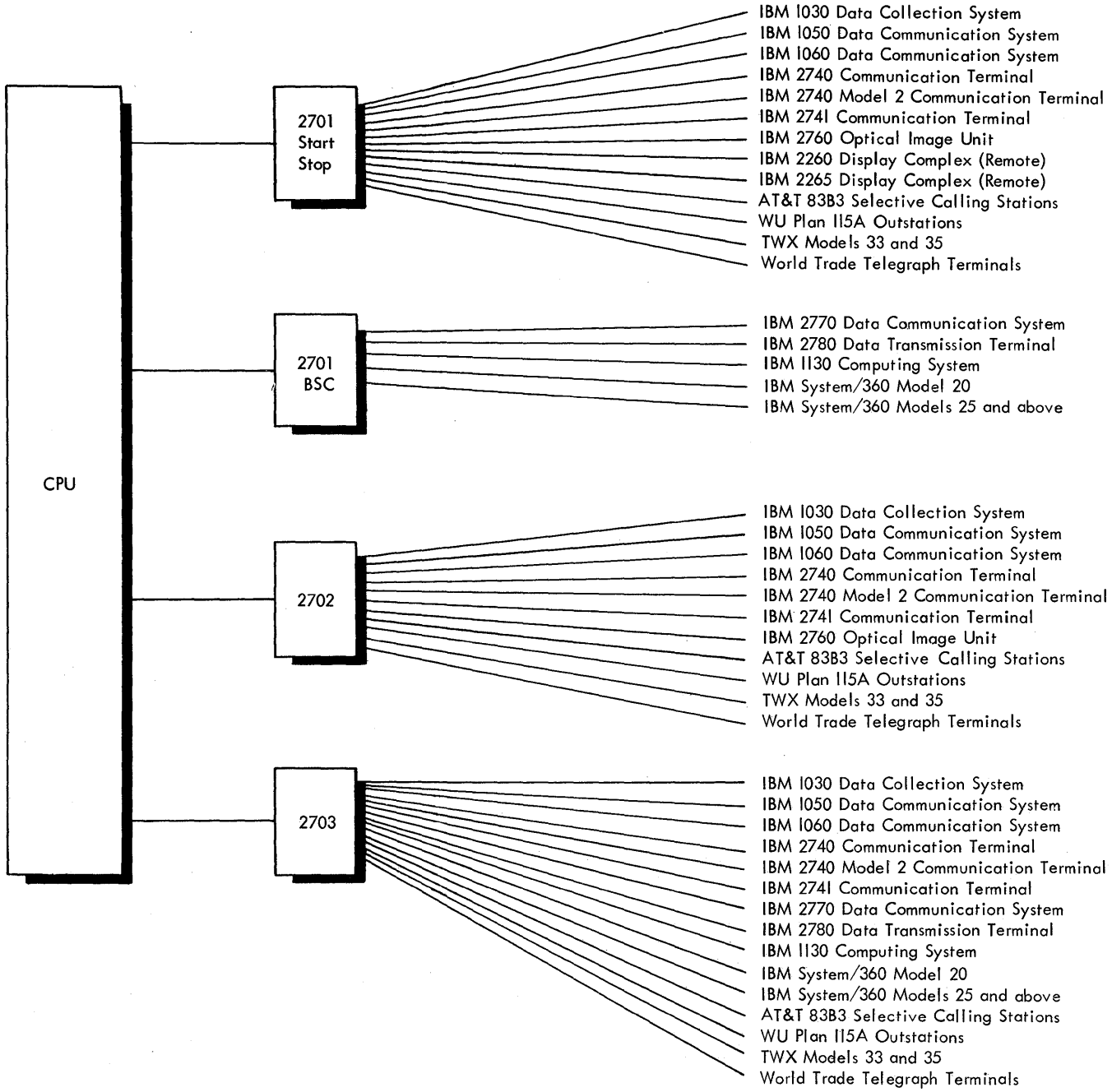

Figure 7. Device Configurations Supported by TCAM (Part 1 of 3)

Station Type		Channel Type		TCU			Audio Response Unit	Line Type		Notes
		Multiplexer	Selector	IBM 2701 Data Adapter Unit	IBM 2702 Transmission Control	IBM 2703 Transmission Control	IBM 7770 Model 3	Switched	Nonswitched	
IBM 1030 Data Collection System	Auto Poll	X			X	X			X	The IBM Digital Time Out feature cannot be attached through an IBM 2701 TCU.
		X		X	X	X			X	
IBM 1050 Data Communication System	Auto Poll	X			X	X			X	
		X		X	X	X		X	X	
IBM 1060 Data Communication System	Auto Poll	X			X	X			X	
		X		X	X	X			X	
IBM 2260-2848 Display Complex (Remote)		X		X					X	
IBM 2260-2848 Display Complex (Local)		X	X							
IBM 2265-2845 Display Complex (Remote)		X		X					X	
IBM 2740 Model 1 Communication Terminal	Auto Poll	X			X	X			X	Two Types: 2740 with station control 2740 with station control and record checking
		X		X	X	X			X	Four Types: 2740 basic 2740 with station control 2740 with record checking 2740 with station control and record checking
		X		X	X	X		X		Four Types, all with dial: 2740 2740 with transmit control 2740 with record checking 2740 with transmit control and record checking
IBM 2740 Model 2 Communication Terminal	Auto Poll	X			X	X			X	Four Types: 2740 2740 with record checking 2740 with buffer receive 2740 without buffer receive (requires line slowdown feature)
		X		X	X	X			X	Four Types: 2740 2740 with record checking 2740 with buffer receive 2740 without buffer receive
IBM 2741 Communication Terminal		X		X	X	X		X	X	The attention feature is not supported, and the break feature is supported only if the CPU is sending and the terminal has not entered data when the break is issued.

Figure 7. Device Configurations Supported by TCAM (Part 2 of 3)

Station Type	Channel Type		TCU			Audio Response Unit	Line Type		Notes
	Multiplexer	Selector	IBM 2701 Data Adapter Unit	IBM 2702 Transmission Control	IBM 2703 Transmission Control	IBM 7770 Model 3	Switched	Nonswitched	
IBM 2760 Optical Image Unit							X	X	Attached to a 2740 Model 1 with record checking
IBM 2770 Data Communication System	X		X		X		X	X	BSC transmission using either ASCII or EBCDIC code
IBM 2780 Data Transmission Terminal	X		X		X		X	X	BSC transmission using ASCII, EBCDIC, or 6-bit code
IBM 1130 Computing System	X		X		X		X	X	BSC transmission
IBM System/360 Model 20	X		X		X		X	X	BSC transmission using either ASCII or EBCDIC code
IBM System/360 Models 25 and above	X		X		X		X	X	BSC transmission and point-to-point lines only
AT&T 83B3 Selective Calling Stations	X		X	X	X			X	
Western Union Plan 115A Outstations	X		X	X	X			X	
TWX Models 33 and 35	X		X	X	X		X		Teletype terminals, dial service (8-level code)
World Trade Telegraph Terminals	X		X	X	X			X	Control unit must incorporate a WTTA
Audio terminals	X					X	X		Example: IBM 2721 Portable Audio Terminal

Figure 7. Device Configurations Supported by TCAM (Part 3 of 3)

Accept: the process in which a destination station acquires a message transmitted to it from the central computer (the message was *sent* by the central computer). Entering and accepting are functions of a station.

Accepting station: a destination station that acquires a message.

Access line: a switched line that continuously connects a remote station and the TCU to a switching center (exchange). A telephone number is associated with the access line.

Access method: a combination of an access technique (either queued or basic) and a given data set organization (for instance, sequential, partitioned, indexed sequential, or direct) that allows the the programmer to transfer data between main storage and I/O devices.

Active line: a communication line that is currently available for transmission of data. Contrast with "inactive line."

Active station: a station that is currently eligible for entering and/or accepting messages on the line. A station may be active for entering or active for accepting, or both, or neither.

Addressing: a non-contention line management method whereby the computer selects remote stations on a multipoint nonswitched line and terminals on a point-to-point line to accept messages sent by the computer.

Addressing characters: identifying characters, sent by the computer, that cause a particular station (or component) to be selected to accept a message.

Allocate: to grant a resource to, or reserve it for, a job or task.

Answer: a procedure by which a called party completes a connection (for switched lines).

Application program: a user-provided program that processes the text portions of messages. Application programs run asynchronously with the Message Control Program, and are usually located in another partition or region of main storage. TCAM application programs are optional; there may be many or none, depending on the needs of the user.

ARU: see "audio response unit."

Assemble: to prepare a machine language program from a symbolic language program by substituting absolute operation codes for symbolic operation codes and absolute or relocatable addresses for symbolic addresses.

Asynchronous operation: message processing being performed in an application program independently from message control being performed in the MCP; asynchronous operations in TCAM result in better utilization of the differences between handling and processing speeds.

Audio inquiry: keying or dialing data into a computer; an IBM 7770 Model 3 Audio Response Unit generates signals that provide an audible response for the person making the inquiry.

Audio line: a communication line attached to an audio response unit such as the IBM 7770 Model 3 Audio Response Unit. An audio communication line is always switched.

Audio response message: an audible response generated by an IBM 7770 Model 3 Audio Response Unit from output it accepts from the computer.

Audio response unit (ARU): a control unit that provides much the same functions for audio stations that a transmission control unit provides for non-audio stations; in addition, it causes an audible response to be made to an audio inquiry.

Audio terminal: a device associated with an IBM 7770 Model 3 Audio Response Unit (ARU), at which keyed or dialed data is entered for transmission to the computer; an audio response is produced by the ARU as output.

Auto answer: a machine feature that allows either a TCU or a station to respond automatically to a call that it receives over a switched line.

Auto Call: a machine feature that allows either a transmission control unit or a station to automatically initiate a call over a switched line. A dialing operation that originates at the central computer must use the Auto Call machine feature.

Auto Poll: A machine feature of a transmission control unit that permits it to handle negative responses to polling without interrupting the central processing unit. At the end of the invitation list, polling is resumed automatically at the beginning of the list.

Available-unit queue: a queue in main storage to which all buffer units are assigned initially (that is, before allocation to TCAM lines and application programs requiring buffers). *Empty* buffer units (that is, buffer units whose contents have been processed by the incoming or outgoing group of an MH, and which are not assigned to the main-storage message queues data set) are returned to the available-unit queue, from which they are reallocated.

Basic access method: any access method in which each input/output statement causes a corresponding machine input/output operation to occur. (The primary macro instructions used are READ and WRITE.)

Batch processing: execution of a sequential set of programs such that each is completed before the next program in the set is started (sometimes referred to as serial execution).

Bid: in the contention method of line management, an attempt by the computer or a station to seize control of the line so that it can transmit data.

Binary synchronous communications (BSC): data transmission in which character synchronization is controlled by timing signals generated by the device that originates a message (and the device that obtains the message recognizes the *sync pattern* at the beginning of the transmission—the devices are locked in step with one another); contrast with *start-stop transmission*.

Block: that portion of a message terminated by an EOB or ETB line control character or, if this is the last block in the message, by an ETX or EOT line control character. When end-of-

block checking is specified in the MH, messages are checked for certain types of transmission and user-specified logical errors on a block-by-block basis.

BSAM: Basic Sequential Access Method.

BSC: see "binary synchronous communications."

Buffer: an area in main storage into which a message segment is read, or from which a message segment is written. Buffers are temporary data-holding areas that are used to compensate for the difference between the rate at which data can be entered from or accepted by a station and the rate at which it can be processed by the CPU; buffers also may be used as work areas in TCAM. The size of TCAM buffers is designated by the user.

Buffer allocation: the assignment of buffers by TCAM to lines or application programs in preparation for reception of message segments from stations on the lines or from application programs. (*See also* "dynamic buffer allocation" *and* "static buffer allocation.")

Buffer deallocation: for a sending operation, deallocation consists of returning the units that comprise the buffer to the available-unit queue after the data in these units has been sent to its destination station or application program; for a receiving operation, deallocation consists of transferring full buffers from the line or application program to which they were assigned to the incoming group of the MH that is to process the message segments they contain.

Buffer prefix: a control area contained within each TCAM buffer. The prefix for the buffer containing the first segment of a message is 30 bytes long, while the prefix for each buffer containing a subsequent segment of the message is 23 bytes long. The user must allow room for the buffer prefix when he specifies his buffer size. TCAM fills the prefix area with buffer control information.

Buffer size: the number of bytes that constitute each buffer in a line group (specified in the DCB macro instruction that defines a line group data set).

Buffer unit: the basic building block from which TCAM buffers are constructed. All units in a particular TCAM system are the same size and are specified by the user.

Buffer unit pool: all the buffer units in a particular TCAM system. The size of the buffer unit pool and the number of units in the pool are specified by the user.

Burst mode: data transfer between main storage and a high-speed I/O device whereby the device monopolizes the I/O interface and stays logically connected to the channel for the transfer of a *burst* of information (the burst may be a few bytes, a whole block of data, or a sequence of blocks with associated control and status information).

Call: a procedure that establishes a connection over a switched line; a series of electrical signals, corresponding to the telephone number of the station or computer with which contact is to be made, are sent down the line; these signals cause automatic switching equipment belonging to the common carrier to establish the connection, if the party being called is free to accept the call.

Cascade entry: an entry in the terminal table associated with a cascade list.

Cascade list: a list of pointers to single, group, or process entries. A message is queued for the valid entry in the list with the fewest messages queued for it.

Catalog: (noun) the collection of all data set indexes maintained by data management; (verb) to include the volume identification of a data set in the catalog.

Central computer: the computer in which TCAM is operating as the control program for the teleprocessing network. Often referred to in this publication as *the computer*.

Central processing unit (CPU): a unit of a computer that includes circuits controlling the interpretation and execution of instructions.

Channel: a mechanical medium for attaching I/O devices to the central processing unit; the channel directs the flow of information between I/O devices and main storage. It relieves the central processing unit of the task of communicating directly with the I/O devices and permits data processing to proceed concurrently with I/O operations (*see also* "selector channel" *and* "multiplexer channel").

Channel program block (CPB): a TCAM control block used in the transfer of the data between buffer units and message queues maintained on disk.

CHECK: a system macro instruction that allows a wait and test for completion of a Read or a Write operation when processing either a sequential or a direct data set.

Checkpoint data set: an optional TCAM data set that contains the checkpoint records used to reconstruct the MCP environment after closedown or system failure, when the TCAM checkpoint/restart facility is utilized.

Checkpoint records: records, located in the checkpoint data set, that are used to reconstruct the MCP environment upon restart following closedown or system failure. The four types of checkpoint records are: environment records, incident records, checkpoint request records, and a control record.

Checkpoint request record: a checkpoint record taken as a result of issuing a special macro instruction in an application program; the record contains the status of a single destination queue for the application program.

Checkpoint/restart: a TCAM facility that records the status of the teleprocessing network at designated intervals or following certain events. Following system failure or closedown, the checkpoint/restart facility uses the records it has taken to restore the MCP environment as nearly as possible to its status before the failure or closedown.

Closedown: an orderly deactivation of the MCP via either an MCPCLOSE macro instruction issued in an application program or an operator command. See "quick closedown" *and* "flush closedown."

Cold restart: start-up of a TCAM MCP following either a flush closedown, a quick closedown, or a system failure; a cold restart ignores the previous environment (that is, the MCP is started as if this were the initial start-up), and is the only type of restart possible when no checkpoint/restart facility is used.

Common carrier: in communications, a government-regulated private company that furnishes the general public with communications services; a telephone or telegraph company.

Communication line: a common-carrier line over which data can be transmitted between telecommunications devices.

Component: an I/O device associated with a station.

Computer: in this publication, the central processing unit in which the TCAM Message Control Program is located.

Concatenated data set: a collection of logically connected data sets (see "job library").

Concurrent: two or more events or activities occurring within the same interval of time.

Contention: classically, a line-control scheme in which stations on a line compete for the use of that line; the station that is successful in seizing control of the line is able to transmit. In a TCAM system, the term is applied to any point-to-point line configuration when the station on the line does not use polling and addressing characters.

Continuation restart: a restart of the TCAM Message Control Program following termination of the MCP due to system failure; the TCAM checkpoint/restart facility is used to restore the MCP environment as nearly as possible to its condition before failure.

Control characters: characters transmitted over a line that are not message data, but which cause certain control operations to be performed when encountered by the computer, transmission control unit, or station; among such operations are polling and addressing, message delimiting and blocking, transmission-error checking, and carriage return.

Control record: a record included in a checkpoint data set that keeps track of the correct environment, incident, and checkpoint request records to use for reconstructing the Message Control Program environment during restart.

Control unit:

1. an integral part of a terminal that uses logic to control I/O operations; for instance, it frees a line when it recognizes an end-of-transmission character, and it recognizes an error condition and forwards a response to the appropriate destination should a printer exhaust its paper supply.
2. a transmission control unit (either an IBM 2701 Data Adapter Unit Model 1, an IBM 2702 Transmission Control Model 1, or a 2703 Transmission Control Model 1) or audio response unit (IBM 7770 Audio Response Unit Model 3); these units provide an interface between communication lines and the computer for logical operations.

CPB: see "channel program block."

CPU: see "central processing unit."

Cross-talk: a phenomenon where one communication line picks up some of the signal that is traveling on another communication line.

DASD: see "direct-access storage device."

Data adapter: an IBM control unit (the IBM 2701 Data Adapter Unit Model 1).

Data collection: a telecommunications application in which data from several locations is accumulated at one location (in a queue or on a file) before batch processing.

Data control block (DCB): an area of main storage that serves as a logical connector between the problem program and a data set. The data control block also can be used to provide control information for any transfer of data. A data control block must be created for each TCAM data set except a message queues data set residing in main storage; a DCB macro instruction is used to create a data control block.

Data link: the communications channel, data sets (or line adapters), and the communications-control portion of each BSC station on a particular channel.

Data set:

1. a named, organized collection of logically related records (program data set). The information is not restricted to a specific type, purpose, or storage medium. Among the data sets specifically related to TCAM are the line group data sets, the message queues data sets, the checkpoint data set, the message log data set, and the input and output data sets for a TCAM-compatible application program.
2. a device containing the electrical circuity necessary to connect data processing equipment to a communication channel; also called a subset, Data-Phone*, modulator/demodulator, or modem.

Delimiter macro instruction: a TCAM macro instruction that classifies and identifies sequences of functional macro instructions and directs control to the appropriate sequence of functional macro instructions.

Destination: the place to which a message is to be sent after it leaves a Message Handler. A destination may be either a station, a group of stations, or an application program. One or more destinations may be specified by the user.

Destination field: a field in a message header containing the name of a station or application program to which a message is directed.

Destination queue: a queue on which messages bound for a particular destination are placed after being processed by the incoming group of a Message Handler. A separate destination queue is created for each station that is queued by terminal, one for each line that is queued by line, and one for each application-program process entry to which the application program may direct GET or READ macro instructions. Destination queues are maintained in message queues data sets which may be located on disk or in main storage. Queuing messages by destination permits overlap of line usage in I/O operations. See *also* "process queue."

Destination station: a station that accepts a message sent to it via the outgoing group of the Message Handler that is specified for the line to which the accepting station is assigned.

Dial: see "call."

Dial line: see "switched line."

Direct-access storage device (DASD): in TCAM, the IBM 2311 Disk Storage Drive and the IBM 2314 Direct Access Storage Facility. Also referred to as *disk*.

---

*Trademark of the American Telephone & Telegraph Co.

Disabling the line: a process whereby TCAM causes the computer to condition either the transmission control unit or the audio response unit to ignore incoming calls on a switched line. Once this is accomplished, the line is available for TCAM to send queued messages to a station on that line. *See* "enabling the line."

Disk: see "direct-access storage device."

Distribution entry: an entry in the terminal table associated with a distribution list.

Distribution list: a list of single, group, cascade or process entries; when a message is directed to the distribution entry associated with this list, TCAM sends the message to each destination named in the list.

Dynamic buffer allocation: the assignment of buffers to a line on an as-needed basis, after a message has started coming in over the line; dynamic allocation occurs following program-controlled interruptions. *See also* "static buffer allocation."

EBCDIC: Extended Binary Coded Decimal Interchange Code.

Enabling the line: a process whereby TCAM causes the computer to condition either the transmission control unit or the audio response unit to respond to incoming calls on a switched line. *See* "disabling the line."

End-of-address (EOA) character:

1. a control character or characters transmitted on a line to indicate the end of non-text characters (for example, addressing characters).
2. a TCAM character that must be placed in a message if the system is to accommodate routing of that message to several destinations; the character must immediately follow the last destination code in the message header.

Enter: the process in which a station places on the line a message to be transmitted to the central computer (the central computer *receives* the message). Entering and accepting are functions of a station.

Environment record: a record of the total teleprocessing environment at a specific moment. The environment record resides in the checkpoint data set; at restart time, an environment record is updated by the contents of incident records that were taken after the environment record was taken, and the updated environment record is then used to reconstruct the Message Control Program environment as it existed before MCP closedown or failure.

EOA: see "end-of-address character."

ERP: see "error recovery procedures."

Error recovery procedures (ERP): a set of internal TCAM routines that attempt to recover from transmission errors.

Exchange: a communications switching center.

FEFO (first-ended first-out): a queuing scheme whereby messages on a destination queue are sent to the destination on a "first-ended first-out" basis within priority groups. That is, higher-priority messages are sent before lower-priority messages; when two messages on a queue have equal priority, the one whose final segment arrived at the queue earliest is sent first.

Flush closedown: a closedown of the TCAM Message Control Program during which incoming message traffic is suspended and queued outgoing messages are sent to their destinations before closedown is completed; this form of termination is known as a *flush* closedown because unsent messages are flushed from the message queues. *See also* "quick closedown."

Functional macro instructions: TCAM macro instructions that perform the specific operations required for messages directed to the Message Handler (see "delimiter macro instructions").

GET: the system macro instruction used in the queued data access technique that obtains a record from an input data set; it operates in a logically-sequential and device-independent manner (*see also* "PUT," "READ," *and* "WRITE").

Group entry: an entry in the terminal table associated with a group of terminals having the group-code machine feature.

Half-duplex: a communication line over which transmissions can occur in either direction, but only in one direction at a time.

Handling: functions provided by the Message Control Program. *See* "Message Handler."

Header: that portion of a message containing control information for the message; a header might contain one or more destination fields, the name of the originating station, an input sequence number, a character string indicating the type of message, a priority level for the message, etc. The message header is operated on by macros in the inheader and outheader subgroups of the Message Handler.

Header buffer: a buffer containing a header segment.

Header segment: a message segment containing all or part of the message header.

Identification characters (ID characters): characters sent by a BSC station on a switched line to identify the station. ID characters can also be assigned to the computer; in this case, the computer and the station can exchange ID sequences. TWX stations also use ID characters.

Identification exchange: a process whereby a switched BSC station calls the computer and then enters its identification sequence (characters) and receives the identification sequence of the computer before message transmission can begin.

Inactive line: a communication line that is not currently available for transmission of data. Contrast with "active line."

Inactive station: a station that is currently ineligible for entering and/or accepting messages. A station may be inactive for entering or inactive for accepting, or both, or neither; the status of a station is determined by the status of the line it is on, by a special character coded in the invitation list entry for the station, by the presence or absence of a functional macro instruction in the outgoing group of the Message Handler handling outgoing messages for this station, and by various operator commands that directly affect the station's status.

Incident record: a checkpoint record residing in the checkpoint data set; an incident record records a change in station status or in the contents of an option field which occurred since the last environment record was taken. Incident records are used

to update the information contained in environment records at restart time after a closedown or system failure.

Incoming: describes messages being transmitted *to* the computer into which TCAM is loaded.

Incoming group: See "Message Handler."

Incoming message: a message being transmitted from a station to the computer. See "outgoing message."

Input: of or related to a message transmission that involves entering data at a station or receiving data at the computer.

Input sequence number: a number placed in the header of each message that is entered by either a station or an application program; this number is checked by a functional macro instruction in the incoming group of a Message Handler to ensure that messages are received from a source in the correct order. The functional macro instruction checks the sequence number for each message; if the number is not one more than that assigned to the previous message received from that origin, the user is made aware that an error has occurred. The use of input sequence numbers is optional.

Inquiry processing: a TCAM application in which the Message Control Program receives a message from a station, then routes it to an application program that processes the data in the message and generates a reply; the reply is routed by the Message Control Program to the inquiring station. Response time often may be shortened by specifying the lock mode (via a functional macro in the Message Handler) and by locating the message queues data set containing the queues for the application program in main storage.

Intercepted station: a station to which no messages may be sent. A station is intercepted by issuing a functional macro instruction in the outmessage subgroup of a Message Handler; the suspension is either for a specified time interval or until either an operator command or an application program macro instruction is issued to release messages held for the intercepted station (an intercepted station can still enter messages—only messages queued *for* the destination are stopped).

Invalid destination: a destination specified for a message that does not correspond to a valid terminal table entry.

Invitation: the process in which the computer contacts a station in order to allow the station to transmit a message if it has one ready.

Invitation delay: a period of time during which outgoing messages are sent to nonswitched polled stations for which receiving has priority over sending. This delay is observed for all polled stations on the line when the end of the invitation list for that line is reached. The delay in polling is observed for such stations whether or not the computer has any messages to send them. If no invitation delay is specified for such stations, no messages can be sent to them. *See also* "system interval."

Invitation List: a series of sets of polling characters or identification sequences associated with the stations on a line; the order in which sets of polling characters are specified determines the order in which polled stations are invited to enter messages on the line.

Inward WATS: a service provided by the telephone company, whereby all incoming calls from within a prescribed area are charged to the receiving subscriber at a flat rate (*see also* "WATS").

Job: a specified group of tasks prescribed as a unit of work for a computer; by extension, a job usually includes all necessary programs, linkages, files, and instructions to the operating system.

Job library: a concatenation of user-defined, partitioned data sets used as the primary source of load modules for a given job.

Line: the communications medium linking the computer to one or more remote stations; message transmission occurs over this medium. (*See also* "nonswitched line," "switched line," "point-to-point line," *and* "multipoint line.")

Line adapter: an IBM data set (modem) that is associated with a particular device.

Line control: the scheme of operating procedures and control signals by which a telecommunications system is controlled.

Line control characters: characters that control transmission of data over a line; line control characters delimit messages, cause transmission-error checking to be performed, indicate whether a station has data to send or is ready to receive data, etc.

Line group: a set of one or more communication lines of the same type, over which stations with similar characteristics can communicate with the computer.

Line group data set: a Message Control Program data set consisting of all the lines in a line group; the messages that are transmitted on these lines constitute the data in this data set. A line group data set is defined by a line group DCB macro instruction, and by a DD statement for each line in the line group.

Line group DCB: a data control block created by a line group DCB macro instruction; information in the data control block defines the line group to TCAM.

Line traffic: data transmission on a line.

Link edit: the combination and editing of object modules to produce a single load module that can be brought into main storage for execution by program fetch. One or more object modules may be linkage edited into a single load module.

Local cable: a cable connected directly to the computer (*not* connected through a transmission control unit or an audio response unit).

Local station: a station whose control unit is connected directly to a computer data channel by a local cable (see "remote station").

Locate mode: a processing mode provided by the queued access technique, and specified in either an input or an output DCB macro instruction, whereby data is processed in a buffer instead of the user work area; this is accomplished by passing the address of the next input or output buffer to the user in a register instead of moving the record into his work area. *See also* "move mode" *and* "substitute mode."

Lock mode: a TCAM facility, invoked in a Message Handler by a functional macro instruction, whereby a station entering an inquiry message for an application program is held on the line by the Message Control Program until a response has been

returned to it by the application program. Use of the lock mode decreases response time because there are no interruptions on the line before a response is returned. A station may be placed in lock mode either for the duration of a single inquiry and response or for the duration of several inquiry-response cycles.

Log: a collection of messages or message segments placed on a secondary storage device for accounting or data collection purposes. The TCAM logging facility is invoked by a functional macro instruction issued in a Message Handler.

Log data set: a data set consisting of the messages or message segments recorded on a secondary storage medium by the TCAM logging facility. A log data set is defined by means of a BSAM DCB macro instruction which is issued with the DCB macro instructions defining the line group data sets, the message queues data sets, and the checkpoint data set.

MCP: see "Message Control Program."

MCPCLOSE: a TCAM macro instruction, issued in the user's application program, that initiates closedown of the TCAM system.

Message: a unit of data received from or sent to a station that is terminated by an EOT or ETX control character or, possibly, by an ETB or EOB control character. A TCAM message is often divided into a header portion, which contains control information, and a text portion, which contains the part of the message of concern to the party ultimately receiving it.

Message Control Program (MCP): a set of user-defined TCAM routines that identify the teleprocessing network to the System/360 Operating System, establish the line control required for the various kinds of stations and modes of connection, and control the handling and routing of messages to fit the user's requirements.

Message Handler (MH): a sequence of user-specified TCAM macro instructions in the Message Control Program that examine and process control information in message headers, and perform functions necessary to prepare message segments for forwarding to their destinations. One Message Handler must be assigned to each line group and one must be assigned to each TCAM-compatible application program. The incoming group of an MH handles messages received from either an originating station or an application program; the outgoing group of an MH handles messages prior to their being sent to a destination station or application program.

Message priority: refers to the order in which messages in a destination queue are transmitted to the destination, relative to each other. Higher-priority messages are forwarded before lower-priority messages. Up to 255 different priority levels may be assigned to a single destination. The priority for each message sent to the destination may be specified in the message header or assigned by a functional macro instruction; in either case, the functional macro instruction should be coded in the inheader subgroup handling the message.

Message queue: see "destination queue."

Message queues data set: a TCAM data set that contains one or more destination queues. A message queues data set contains messages that have been processed by the incoming group of a Message Handler and are waiting for TCAM to dequeue them, route them through an outgoing group of a Message Handler, and send them to their destinations. Up to three message queues data sets may be specified for a TCAM Message Control Program.

Message segment: the portion of a message contained in a single buffer.

Message switching: a telecommunications application in which a message is received from a remote station, stored until a suitable outgoing line is available, and then transmittted to its destination station. TCAM message switching can be handled entirely by the Message Control Program.

MFT-II: multiprogramming with a fixed number of tasks; an Operating System control program option that supervises the execution of more than one job at a time. TCAM may operate in an MFT-II environment. See *also* "MVT."

MH: see "Message Handler."

Modem: see "data set."

Move mode: a processing mode provided by the queued access technique and specified in either an input or an output DCB macro instruction, whereby data is processed in a user work area; this is accomplished by moving the record that is to be processed into the user work area. See *also* "locate mode" *and* "substitute mode."

Multiplexer channel: a channel that provides the capability of two or more channels by either splitting the frequency band transmitted over the channel into narrower bands, each of which is used as a distinct channel (frequency-division multiplexing), or allocating this common channel to several different information channels, one at a time (time-division multiplexing).

Multipoint line: a nonswitched line that connects several remote stations to the computer.

MVT: multiprogramming with a variable number of tasks; MVT, like MFT-II, supervises execution of more than one job step at a time. In addition it allocates main storage dynamically to each job. A TCAM MVT environment supports the large job customer and the customer who has many small jobs. See *also* "MFT-II."

Nonswitched line: a communication line that links stations for a continuous period, or for regularly recurring periods; also known as a private, leased, or dedicated line.

Nonswitched network: a group of nonswitched lines that connect the computer with remote stations.

Nontransparent mode: a mode of BSC transmission in which all control characters are treated as control characters (that is, not treated as text). See "transparent mode"

NOP: an instruction that specifically instructs the computer to proceed to the next instruction in sequence. In TCAM, certain QTAM macro instructions (COPYP, COPYT, CHNGP, CHNGT) are each assembled as a NOP; their corresponding functions are provided either by issuing appropriate TCAM macro instructions, or by entering operator commands.

On-line test: an optional TCAM facility that permits either a system console operator or a remote-station operator to test transmission control units and remote stations to find out if they work properly.

Open list: an invitation list that is polled station-by-station through the list; at the end of the list, a prespecified polling delay is observed before the next polling pass through the list (see "wrap list").

Operator command: a command entered either at an operator control station, at the system console, or from an application program to examine or alter the status of the telecommunications network during execution.

Operator control station: a station eligible to enter operator commands. An application program or the system console may also serve as operator control stations. *See also* "primary operator control station" *and* "secondary operator control station."

Option field: a storage area containing data relating to a particular station, component, line, or application program; certain Message Handler routines that need source- or destination-related data to perform their functions have access to data in an option field. User-written routines also have access to data in an option field.

Originating station: a station from which a message, or other data, originates.

Outgoing message: describes messages being transmitted *from* the computer to a destination.

Outgoing group: See "Message Handler."

Output: message transmission that involves either accepting data at a station, or sending data from the computer.

Output sequence number: a number placed in the header of a message by TCAM that determines the order in which messages were sent to a destination by the computer. When specified in an outheader subgroup, a functional macro instruction causes an output sequence number to be placed in the header of each outgoing message; this sequence number is one greater than the sequence number for the last message sent to this destination. *See also* "input sequence number."

Partition: a discrete subdivision of the dynamic area in main storage in an MFT-II environment. Main storage is divided into two major areas: the system area (reserved for control program modules) and the dynamic area. The dynamic area is divided into a number of discrete areas called partitions. *See also* "region."

PCI: *see* "program-controlled interruption."

POINT: a system macro instruction that, when issued in a TCAM environment in conjunction with either a GET or a READ macro instruction, identifies a station by passing in a register or a field the station identification and the sequence number of a message to be retrieved (this macro instruction is used to retrieve a message that has already been placed on a DASD destination queue).

Point-to-point line: a communication line that connects a single remote station to the computer. It may be either switched or nonswitched.

Polling: a non-contention line management method whereby the computer invites remote stations on multipoint nonswitched lines and remote terminals on point-to-point lines to enter messages. The computer contacts stations in the order specified by the invitation list; each station contacted is invited to enter messages.

Polling characters: a set of identifying characters peculiar to either a station or a component of that station; a response to these characters indicates to the computer whether the station has a message to enter.

Polling delay: a user-specified delay between polling passes through an invitation list for either a line or a line group. *See also* "system interval."

Polling pass: a system in which the computer invites each station, in a specified sequential order, to enter any messages it has ready.

Prefix: *see* "buffer prefix."

Primary operator control station: the operator control station that can receive an error recovery procedure message and send operator commands and receive related responses. *See also* "secondary operator control station."

Primary storage: main storage.

Priority: *see* "message priority" *and* "transmission priority."

Process queue: a destination queue for an application program.

Processing: functions provided by a user-designed application program whereby messages are acted upon to fit the needs of the user's application.

Program data set: *see* "data set."

Program-controlled interruption (PCI): an interruption, specified in the line group DCB macro instruction, that allows buffers to be deallocated continuously, thus replenishing the free-unit pool.

PUT: a system macro instruction used in the queued data access technique that places a record in an output data set; it operates in a logically-sequential and device-independent manner (*see also* "GET," "READ," *and* "WRITE").

QSAM: Queued Sequential Access Method.

QSTART: a TCAM macro instruction that differentiates between QTAM and TCAM application programs when QTAM application programs are being run in a TCAM environment.

Queue: a set of items consisting of:

1. a queue control block (an area in main storage containing control information for the queue), and
2. one or more ordered arrangements of items (the items may be messages, main storage addresses, etc.).

Queued access method: any access method that automatically synchronizes the transfer of data between the program using the access method and input/output devices, thereby eliminating delays for input/output operations.

Quick closedown: a closedown of the TCAM Message Control Program that entails stopping message traffic on each line as soon as any messages being sent or received at the time the request for closedown is received are transmitted.

READ: a system macro instruction in the basic data access technique that retrieves a data block from an input data set and places it in a user-designated work area in main storage; the basic data access technique is used when the sequence in which the records are to be processed is unknown, or when the user does not want some or all of the automatic functions performed by the queued access technique. *See also* "WRITE," "GET," *and* "PUT."

Read-ahead queue: an area of main storage from which an application program obtains work units, using GET or READ macro instructions, to move them to a work area.

Receive: the process in which the central computer obtains a message from a remote station (the message is *entered* by the station). Receiving and sending are functions of the central computer.

Record: a unit of data being moved into an application program work area; the work area terminates when :

1. the work area has been filled, or
2. the length designated by an operand of the input DCB macro instruction has been reached (fixed-length records only), or
3. a user-specified delimiting character is encountered (specified by an operand of a TCAM editing macro in the Message Control Program).

Records may be of fixed, variable, or undefined length.

Region: a discrete subdivision of the dynamic area of main storage in an MVT environment. Main storage is divided into two major areas: the system area (reserved for control program modules) and the dynamic area. The dynamic area is divided into regions. Each job initiated operates in a region and up to 15 independent jobs can be running concurrently. *See also* "partition."

Remote station: a station that is connected to a computer data channel through either a transmission control unit or an audio response unit. *See also* "local station."

Reserve characters: characters inserted in a message header by TCAM to reserve room in the buffer for later insertion of the date, time, and sequence number of a message, and for the screen control character for the IBM 2260 and 2265 Display Stations.

Resource: any facility of the computing system or operating system required by a job or task, including main storage, I/O devices, the central processing unit, data sets, and control processing programs.

Restart: any TCAM start-up (cold restart, warm restart, and continuation restart) other than the initial start-up.

SAM: *see* "sequential access method."

Secondary operator control station: an operator control station that can send operator commands and receive related responses. *See also* "primary operator control station."

Secondary storage: Auxiliary storage (not main storage).

Segment: the portion of a TCAM message contained in a single buffer.

Seize: to gain control of a line for transmitting data. *See also* "bid."

Selection: the process whereby the computer contacts a remote station to send it a message.

Selector channel: a communication channel that operates only in burst mode for handling data transfer to and from high-speed I/O devices.

Send: the process in which the central computer places a message on a line for transmission to a station (the station *accepts* the message). Sending and receiving are functions of the central computer.

Sequence number: *see* "input sequence number" *and* "output sequence number."

Sequential access method (SAM): either QSAM (using GET and PUT macro instructions for data transfer to and from the Message Control Program) or BSAM (using READ and WRITE macro instructions for data transfer to and from the Message Control Program).

Single entry: an entry in the terminal table associated with a single station or station component; one such entry must be created for each station in the TCAM system not defined by a group entry.

SIO: start I/O. A system control macro instruction that instructs the central processing unit to initiate I/O operations.

Start-stop transmission: data transmission in which each character being transmitted is preceded by a special control signal indicating the beginning of the sequence of data bits representing the character, and is followed by another control signal indicating the end of the data-bit sequence (character recognition by the device that obtains the data depends on the presence of these control signals for each character); *contrast with* "binary synchronous communications."

Static buffer allocation: the assignment to a line, before transmission over that line, of all buffers to be used to contain the transmitted data. When the appropriate operand is coded in the line group DCB macro instruction, the number of buffers specified by another operand of the line group DCB macro instruction is assigned to a line before incoming or outgoing transmission begins on that line; once transmission has started, no more buffers are available to handle the data involved in the transmission.

Station: either a remote terminal, or a remote computer used as a terminal.

Sub-block: that portion of a BSC message terminated by an ITB line control character.

Subgroup: a subdivision of either an incoming group or an outgoing group. An incoming group comprises inheader, inbuffer, and inmessage subgroups; an outgoing group comprises outheader, outbuffer, and outmessage subgroups.

Substitute mode: a processing mode provided by the queued data access technique, and specified in either an input or an output DCB macro instruction, whereby the address of the next input or output buffer is interchanged with the address of the user work area instead of moving a record (*see also* "locate mode" *and* "move mode").

Switched line: a communication line on which the connection between the computer and a remote station is established by dialing. Also known as a dial line.

Switched network: a group of switched lines that connects the computer with remote stations.

System closedown: *see* "closedown."

System interval: a user-specified time interval during which polling and addressing are suspended on multipoint lines to polled stations. The system interval is specified during TCAM initialization and is initiated subsequently by an operator command; it optimizes productive polling by minimizing central

processing unit meter time and it synchronizes polling on the polled lines in the system. The duration of subsequent system intervals can be changed with another operator command. See *also* "invitation delay."

Task: the smallest independent unit of work that can compete for the resources of the computing system.

TCU: see "transmission control unit."

Telecommunications: any transmission or reception of signals, writing, sounds, or intelligence of any nature, by wire, radio, or other electromagnetic media.

Teleprocessing: the processing by a computer of data entered at a remote station.

Terminal: a point in a system at which data can enter, leave, or enter and leave. A terminal can also be a control unit to which one or more input/output devices can be attached (see "component").

Terminal table: an ordered collection of information consisting of a control field for the table and blocks of information on each line, station, component, or application program from which a message can originate or to which a message can be sent.

Termination: see "closedown."

Text: that part of the message of concern to the party ultimately receiving the message (that is, the message exclusive of the header, or control, information).

Text processing: a function whereby a user-specified (and designed) application program processes the text portion of a message.

Transmission: the transfer of coded data via an electromagnetic medium between two points in a telecommunications network.

Transmission code: the physical scheme of representing data as it is being transmitted via common-carrier facilities.

Transmission control unit (TCU): a control unit that serves as an interface between communication lines and a computer for logical operations. The transmission control units supported by TCAM are the 2701 Data Adapter Unit Model 1, the 2702 Transmission Control Model 1, and the 2703 Transmission Control Model 1.

Transmission priority: refers to the order in which sending and receiving occur, relative to each other, for a particular station. Transmission priority is specified on a line-group basis by an operand of the line group DCB macro instruction. The three transmission priorities possible in TCAM are send priority, equal priority, and receive priority. The exact meaning of each priority depends upon the line configuration and type of station. (See *also* "message priority.")

Transmission technique: the method used to manage the transfer of data over a communication line. See "binary synchronous communications" and "start-stop transmission."

Transmit: see "transmission."

Transparent mode: a mode of BSC transmission in which all data, including normally restricted data-link control characters, is transmitted only as specific bit patterns. Control characters that are intended to be effective are preceded by a DLE character.

TWX: abbreviation of Teletypewriter Exchange Service, a semi-automatic switching service provided by AT & T for interconnecting public teletypewriter subscribers.

Unit: see "buffer unit *or* "work unit."

Warm restart: a restart of the TCAM Message Control Program following either a quick or a flush closedown; the TCAM checkpoint/restart facility is used to restore the MCP environment as nearly as possible to its condition before failure. See "restart."

WATS: abbreviation for AT & T's Wide Area Telephone Service, which provides a special line on which the subscriber may make unlimited calls to certain zones on a direct distance dialing basis for a flat monthly charge.

Work area: an area designated for an application program that is used for handling the work units obtained by a GET or READ macro instruction.

Work unit: the unit of data that is obtained by a GET or READ macro instruction for processing in an application program. The work unit may be a message or a record (or, for QTAM-compatible application programs, a segment).

Wrap list: an invitation list that is polled station-by-station through the list. At the end of the list, polling resumes without delay at the beginning of the list (that is, the invitation list is treated as though it has no end). See *also* "open list."

WRITE: a system macro instruction used in the basic data access technique that places a data block in an output data set from a designated work area in main storage. (See *also* "READ," "GET," and "PUT.")

WTO (write to operator): an optional user-coded service whereby a message may be written to the system console operator informing him of errors and unusual system conditions that may need correcting.

READER'S COMMENT FORM

IBM System/360 Operating System
Telecommunication Access Method (TCAM)
Concepts and Facilities

Order No. GC30-2022-0

- How did you use this publication?

  As a reference source ☐
  As a classroom text ☐
  As a self-study text ☐

- Based on your own experience, rate this publication . . .

As a reference source:	Very Good	Good	Fair	Poor	Very Poor
As a text:	Very Good	Good	Fair	Poor	Very Poor

- What is your occupation?

- We would appreciate your other comments; please give specific page and line references where appropriate. If you wish a reply, be sure to include your name and address.

- Thank you for your cooperation. No postage necessary if mailed in the U.S.A.

YOUR COMMENTS, PLEASE . . .

This publication is one of a series that serves as a reference source for systems analysts, programmers, and operators of IBM systems. Your answers to the questions on the back of this form, together with your comments, help us produce better publications for your use. Each reply is carefully reviewed by the persons responsible for writing and publishing this material. All comments and suggestions become the property of IBM.

Please note: Requests for copies of publications and for assistance in using your IBM system should be directed to your IBM representative or to the IBM sales office serving your locality.

Fold Fold

FIRST CLASS
PERMIT NO. 569
RESEARCH TRIANGLE PARK
NORTH CAROLINA

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U. S. A.

POSTAGE WILL BE PAID BY . . .

IBM Corporation
P. O. Box 12275
Research Triangle Park
North Carolina 27709

Attention: Publications Center, Dept. E01

Fold Fold

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)

Cut Along Line

IBM S/360 OS TCAM Concepts and Facilities (S360-30) Printed in U.S.A. GC30-2022-0

GC30-2022-0

IBM S/360 OS TCAM Concepts and Facilities (S360-30) Printed in U.S.A. GC30-2022-0

IBM

International Business Machines Corporation
Data Processing Division
1133 Westchester Avenue, White Plains, New York 10604
(U.S.A. only)

IBM World Trade Corporation
821 United Nations Plaza, New York, New York 10017
(International)