THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 62, No. 7, September 1983
Printed in U.S.A.

Total Network Data System:

Equipment Systems

By N. D. FULTON,* J. J. GALIARDI,* E. J. PASTERNAK,*
S. A. SCHULMAN,* and H. E. VOIGT*

(Manuscript received March 10, 1983)

This article describes a set of five systems collectively called the Total Network Data System Equipment Systems, which support a number of network administration and engineering functions in the operating telephone companies. These systems are run by the operating telephone companies on large mainframe computers. In an average telephone company they support the management of the collection and distribution of over 50 million individual items of network data each week. They also provide reports to assist in the engineering and administration of No. 5 Crossbar switching offices, and the load balancing of all local switching systems. Initial development of these systems occurred in the early to mid-1970s. They are installed in all Bell System operating telephone companies and Bell Canada. Currently, on-line record base updates and inquiries and on-line viewing of report outputs are being developed.

I. INTRODUCTION

1.1 Role of TNDS/EQ in support of network functions

Basic to the operation of the telephone business is the collection, processing, distribution, and analysis of traffic data. The usage and availability for service of each of the many millions of components in

* Bell Laboratories.

the telephone system are measured by evaluating these data. In a typical week, the average Operating Telephone Company (OTC) collects over 50 million separate traffic measurements. Aided by the reports formulated from these data, telephone company personnel monitor how well customers are being served, place work orders as needed to optimize usage of currently installed equipment, and formulate traffic trends to forecast future needs. Almost all major decisions in the managing of the telephone network are influenced by the conclusions derived from traffic data.

The group of systems referred to as the TNDS Equipment (TNDS/EQ) systems primarily support data administration and central office switching engineering and administration functions in network organizations. Included in data administration is the collection, verification, and delivery of traffic data, both to the network and to other users. (As an example of the latter, local/toll call separations data are delivered to Division of Revenues.) One function of central office switching engineering and administration is to monitor usage patterns of the many individual traffic-sensitive components in a central office to ensure good customer service along with proper equipment utilization.

1.2 Organization of article

The remainder of Section I provides an overview of TNDS/EQ and discusses some of the history and decisions involved in its evolution. Section II describes the data collection environment, which serves as the input to the TNDS/EQ systems. Section III discusses the computer operational environment under which these systems run, and describes some common controls that were initiated to improve this environment. Sections IV through X describe in detail each of the TNDS/EQ components, and Section XI indicates some of the future directions for TNDS/EQ.

1.3 Overview of TNDS/EQ

TNDS/EQ comprises five component systems (see Fig. 1) and two special facilities. As discussed in Section 1.4, planning for these systems goes back to the mid-1960s with most initial development occurring in the early to mid-1970s. They are currently used throughout all Bell System operating telephone companies and Bell Canada. TNDS/EQ operates systems on local IBM 370 or equivalent mainframe computers, and are run weekly in a batch processing mode. Comprising over one-half million instructions, the programs have been written in a combination of COBOL, FORTRAN, PL/1, and Basic Assembly Language. Products delivered to the operating companies include executable load modules, IBM job control language, installa-

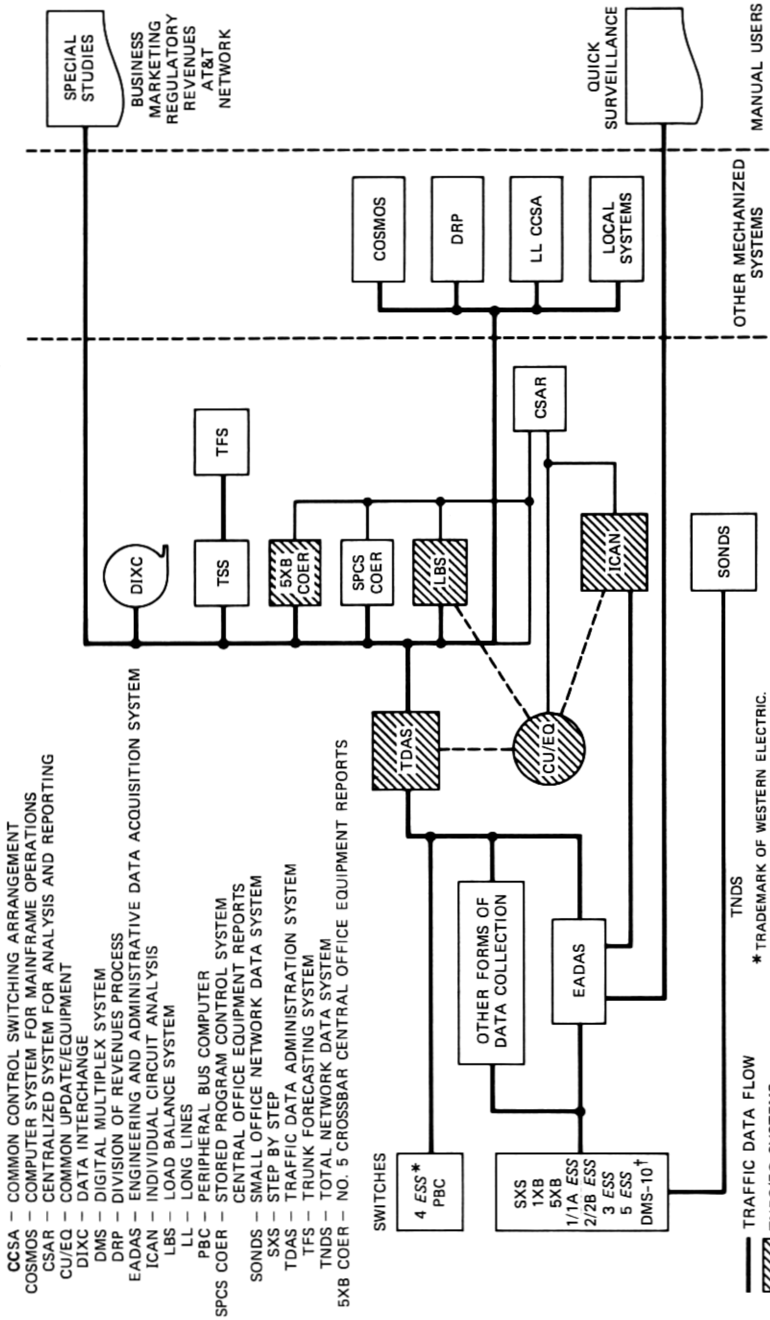


Fig. 1—TNDS data processing.

tion and conversion instructions, and documentation to support both computer center operations personnel and network users.

Three other systems are sometimes considered part of TNDS/EQ—the Stored Program Control System Central Office Equipment Reports System (SPCS COER),[1] the Centralized System for Analysis and Reporting (CSAR),[2] and the Small Office Network Data System (SONDS).[3] Because of their different operating environment—running on a single, time-shared computer for all using companies, rather than being deployed to each operating telephone company—they are presented in other articles in this issue.

One of the TNDS/EQ components, Common Update (CU), maintains master files used by a number of the other TNDS component systems (see Section IV). These files contain the basic records needed to process the data. As an example, the CU master files contain information that defines what data are to be transmitted to each of the downstream systems.

The Traffic Data Administration System (TDAS) is the key TNDS element serving the data administration functions in the operating companies (see Section V). TDAS accepts traffic data from most data collection sources, including paper tape, punched card, and magnetic tape. Among the sources of data are the Western Electric Engineering and Administrative Data Acquisition System (EADAS),[4] several non-Bell System data acquisition systems, and those switching systems, such as the Bell System 4 *ESS** switching equipment, which internally summarize traffic data. Using CU files, TDAS identifies, verifies, and labels each individual incoming data item, saving those items wanted for further processing and discarding those that are not wanted. On a weekly basis, scheduled data are transmitted to the requesting downstream systems. These output (downstream) interfaces are unlimited, as TDAS can effectively fill any order for data provided the data have been collected and delivered to it.

Included among the downstream systems are two component systems of TNDS/EQ: the Load Balance System [(LBS), see Section VIII], and the No. 5 Crossbar Central Office Equipment Reports System [(5XB COER, see Section VI]. LBS reports assist the network administrator in assigning customer lines to the switching machines in a manner that balances the traffic load on the switch. It also provides telephone company management personnel with quantified measures (known as the Load Balance Index) on how well the office balance is being maintained. 5XB COER reports on weekly, monthly, and annual traffic load for each individual component in a No. 5 Crossbar switching office.

---

* Trademark of Western Electric.

Another component of TNDS/EQ, the Individual Circuit Analysis System (ICAN), serves two basic functions (see Section VII). First, it provides reports to help administer the record base for the Individual Circuit Usage Recording (ICUR) option of EADAS; second, it analyzes individual component usage data to identify unusual measurements or patterns that could be indicative of central office or trunk maintenance problems.

In addition to these five component systems, TNDS/EQ has two special facilities, the Report Distributor (Section IX) and the Management Reporting System [(MRS) see Section X]. The former helps to deliver outputs to the network users and produces both paper and microfiche output. The MRS enables an operating company to access selected portions of the record base and output files to produce its own specialized reports.

1.4 History and major decisions

Elements of TNDS/EQ originated in several different development organizations within Bell Laboratories, AT&T and the Operating Telephone Companies. Most of the components were developed prior to the overall concept of a coordinated Total Network Data System. As a result, many activities over the past decade have been directed at integrating the systems. Some of this history has been addressed in an earlier article.[5] Even though there originally was not a complete plan, the TNDS architecture quickly and naturally fell into an "upstream" and "downstream" split. Placed upstream were the functions associated with data collection and real-time exception reporting, activities suitably accomplished with minicomputer technology. On the other hand, the heavy processing functions involved with crunching tens to hundreds of millions of weekly data items into usable engineering and administrative reports readily fell into a downstream processing environment of large mainframe batch computers.

The history of downstream TNDS components goes back to 1966 (see discussion in Ref. 5), with a decision to centralize the development of a computerized Trunk Facilities System. This decision led to a major development program in the Business Information Systems (BIS) area of Bell Laboratories. Included in the program were three systems that became part of TNDS—The Trunk Servicing System (TSS), The Trunk Forecasting System (TFS), and TDAS. The former two are now part of the Trunking System (TNDS/TK).[6] These three systems, together with Common Update as a supporting record base system, became known as Servicing and Estimating, under Bell Laboratories BIS development. Around the time Servicing and Estimating got its start, a stand-alone reporting system for No. 5 Crossbar central office data was developed by New Jersey Bell. It was standardized and integrated into the TNDS downstream structure by AT&T and a mechanized interface to TDAS was implemented in 1972.

One of the earliest and perhaps most important architectural decisions was the placement of TDAS as a single point of entry for all data to the "downstream" world. In this position, TDAS served as a data warehouse—receiving data on its input side from many different data collection processes and distributing data on its output side to the many different users (both individuals as well as other systems). This basic architecture, established when most data collection was manually transposed and keypunched from Traffic Usage Recorder film outputs, has been extremely robust under the stress of rapidly changing data collection methods and data processing needs. On the output side, new downstream processing systems have been able to satisfy their needs for traffic data by simply placing orders, known as Traffic Measurement Requests (TMRs), on the TDAS data warehouse. And on the input side, new collection systems have been able to deliver their data by defining interfaces to TDAS.

In the mid-1970s, ICAN and LBS were developed and added to the TNDS downstream systems. ICAN was developed by the Network Planning area of Bell Laboratories, while LBS was developed by the BTL Servicing and Estimating organization. Shortly after ICAN was developed, all TNDS downstream systems (Equipment and Trunking) migrated to the Bell Laboratories BIS organization. By 1977 5XB COER, ICAN, and CSAR (which had also been developed by Bell Laboratories Network Planning) had been moved. Around this time, because of the growing size of the development organizations, the downstream systems were split into two BIS departments, along the lines of network user communities (trunking and equipment), to form TNDS/TK and TNDS/EQ as separate system groupings.

Combining all the downstream TNDS systems (TNDS/EQ and TNDS/TK) into one organization enabled common standards and features to be established including standard run control procedures (discussed in Section III), and a report distribution facility (discussed in Section IX).

With the expansion in the mid-1970s of both the number of systems as well as the number of operating telephone company installations, it became necessary to provide strong controls over the release of program modifications—both fixes and new features. Coordination was necessary from many standpoints. For developers, the incorporation of a new feature often required simultaneous changes to several systems. For system maintainers, keeping track of each of 20 operating company configurations of approximately 200 program modules required mechanized aids. And for each operating company, where hundreds of individuals depended on TNDS/EQ outputs to carry out their daily jobs, there was a need to plan for new features and to be kept informed of the status of fixes to problems.

To cope with these needs that are common to large systems developments and large user communities, three major changes were instituted:

1. A coordinated annual release was established for all TNDS/EQ systems for packaging of new features. The scheduling of the release was set to occur during periods of slack telephone traffic so that any disruptions resulting from installing the new release would not interfere with processing busy season data.

2. A development environment was established based on the *UNIX** operating system, utilizing the Source Code Control System (SCCS) and Modification Request Control System (MRCS). These provided management control over different program versions and provided status tracking of trouble reports (modification requests).

3. A functional organization was established with separate requirements, development, test, installation, and consultation groups. Along with this organization, standard procedures and schedules were established for internal development activities. Also, greater emphasis was placed on formal documentation, both to support the internal functional organization and to support telephone company personnel in their planning and training for the annual TNDS/EQ releases.

Starting in the late 1970s attention was directed to the need for modernization of TNDS/EQ, in line with technological advances that had occurred over the past decade. Of all the decisions made, this one was the most difficult. Over the years, the major design focus for TNDS/EQ had been to keep up with network changes, to improve the efficiency of its heavy-duty data processing activities, and to streamline the flow of data from data collection systems, through TDAS to the downstream processing programs. For those TNDS/EQ functions associated with this processing of traffic measurement data, batch processing remained appropriate, even with current state-of-the-art computers.

Batch operations were awkward in areas involving direct user interfaces: inputs to update the record base (Common Update), and deliveries of output reports. On the input side, Common Update's weekly processing cycle introduced long delays in making record base changes and verifying their correctness. When errors occurred, several weeks could elapse before they would be corrected. On the output side, since the batch process does not retain outputs, telephone company personnel would often obtain output volumes far exceeding their needs "just in case" some information might be needed at a later date. It is evident

* Trademark of Bell Laboratories.

that an on-line, interactive interface with the user would significantly improve the environment. But in spite of these opportunities for improving human interfaces, net benefits were difficult to quantify. On the one hand, costs for terminals, printers, telecommunications lines, and computer on-line processing are easily quantified and generally are expensive. On the other hand, savings gained in personnel time from improved input error processing and less paper shuffling are very subjective.

Several activities were conducted to obtain a better view of benefits. In 1980, time-motion studies were conducted at two operating companies to obtain a view on how people, particularly clerical personnel, spent their time. The results of these two studies were consistent and showed that there were significant opportunities for savings through modernization. As a result of these studies, an experimental on-line system was installed in the second half of 1981 at a Chesapeake & Potomac Telephone Company network location and remained in place for about one year. Even though limited in scope, the on-line capabilities were received enthusiastically by the C & P Company personnel, and comparative time-motion studies conducted both before and during the experiment again showed that there were opportunities for savings. As a final activity, each Bell Operating Company (BOC) was requested to conduct its own cost-benefit study to determine whether on-line features would be economically beneficial. These studies were conducted during the summer of 1982 and resulted in the decision to proceed with an on-line system for record base inputs and output report viewing. This system has been named On-line Records and Reporting System (ORRS). The current plan is to provide these capabilities over the 1983 to 1986 time period, with first operating company application occurring in 1983.

II. DATA COLLECTION

Because TNDS/EQ (specifically TDAS) needs to interface with multiple sources of data, a short discussion on data collection is presented here. Methods of data collection as well as types of data collected have changed greatly over the years. TNDS/EQ serves the vital function in the overall TNDS design of translating these multiple inputs to a common output format and identifying the data in terms of Bell System Common Language. The full list of measurement types supported by TNDS/EQ is given in Table I and were discussed in an earlier article.[7]

There was a need to collect telephone traffic data well before there were mechanized systems for collecting the data. In the early 1900s operators who manually completed calls recorded data on the frequency of calls to specific locations. This information was used for

Table I—TNDS/EQ measurement types

Measurement Type	Description	Switching Machines Applicable
PC	Peg count	All
USG	Usage	All
OVF	Overflow	All
MTU	Maintenance usage	1E, 2E, 5XB, 1XB, XBT
INU	Incoming usage	1 *ESS*, 4A, 2 *ESS*
ATB	All trunks busy	SXS
LTB	Last trunk busy	SXS
CMP	Completions	SXS, 5XB, XBT, 1XB
IPC	Incoming peg count	4E, DMS
MBC	Maintenance busy count	2E, 3E
DGU	Detector group usage	SXS, 5XB, XBT, 1XB
RTS	Reroute to seizure	4E

establishing new routes between cities and resizing existing trunk groups. With the development of electromechanical switching systems, the collection of counts was mechanized. Counts were accumulated on mechanical registers. Periodically, the register readings were recorded by clerical personnel and compared with earlier readings to obtain a difference count. The manual recording of traffic register values was later replaced by automatically photographing these registers at predefined intervals.

During the 1960s, the rapid introduction of computer technology brought about three major changes in the data collection process. First, general-purpose computers were used to calculate the register differences. Second, the camera-register method began to be replaced by centralized data collection hardware, designed to collect measurements directly from the switching machines and to record the counts on magnetic tape for later processing by general-purpose computers. The first system of this type was the Traffic Data Recording System (TDRS). This system, implemented as a hard-wired computer, was soon found to lack the flexibility and capacity of the rapidly evolving minicomputer technology. The third major change in the 1960s was the development of electronic switching machines, which could perform their own internal data collection.

During the 1970s, the collection of real-time data introduced new capabilities to support office maintenance and network management. Also during this period, various support systems were centrally developed to run on general-purpose computers, providing features and flexibility applicable to all companies. These centrally developed systems standardized the format and labeling of traffic data so operating telephone companies could conveniently interchange common data.

As a result of this evolutionary process, data are collected many different ways in the Bell System. Table II lists the 35 different collection types currently supported. One of the major ongoing activ-

Table II—Traffic data input formats

Data Collection Type	Description	Switching Machines Supported
R	Register readings	Electromechanical
D	Register differences	Electromechanical
S	Minicomputer scanned data	Electromechanical
A	Minicomputer accumulated data	Electromechanical
P	Minicomputer polled data	Electromechanical
I	Minicomputer individual circuit data	Electromechanical
X-A	4A trunking data	4A
X-D	4A central office data	4A
1-C	1/1A *ESS* switch continuous data	1/1A *ESS*
1-H	1/1A *ESS* switch hourly data	1/1A *ESS*
1-W	1/1A *ESS* switch weekly data	1/1A *ESS*
1-D	1/1A *ESS* switch daily data	1/1A *ESS*
1-T	1/1A *ESS* switch traffic separations data	1/1A *ESS*
1-E	1/1A *ESS* switch extreme value data	1/1A *ESS*
2-C	2/2B *ESS* switch continuous data	2/2B
2-H	2/2B *ESS* switch hourly data	2/2B
2-W	2/2B *ESS* switch weekly data	2/2B
2-D	2/2B *ESS* switch daily data	2/2B
2-R	2B *ESS* switch record verification data	2B
2-E	2B *ESS* switch extreme value data	2B
3-C	3 *ESS* switch continuous data	3 *ESS*
3-H	3 *ESS* switch hourly data	3 *ESS*
3-W	3 *ESS* switch weekly data	3 *ESS*
3-D	3 *ESS* switch daily data	3 *ESS*
4-5	4 *ESS* switch trunking data	4 *ESS*
5-C	5 *ESS* switch continuous data	5 *ESS*
5-H	5 *ESS* switch hourly data	5 *ESS*
5-D	5 *ESS* switch daily data	5 *ESS*
5-R	5 *ESS* switch record verification data	5 *ESS*
5-E	5 *ESS* switch extreme value data	5 *ESS*
M-C	DMS-10 continuous data	DMS-10*
M-H	DMS-10 hourly data	DMS-10
M-D	DMS-10 daily data	DMS-10
M-E	DMS-10 extreme value data	DMS-10
J-C	DMS-200 continuous data	DMS-200

* Trademark of Northern Telecom Inc.

ities of TNDS/EQ is to keep current with the collection environment as it continues to evolve.

III. OPERATIONAL ENVIRONMENT AND CONTROLS

3.1 Introduction

When TNDS/EQ and TNDS/TK were introduced in the early 1970s, the OTC computer operating environment was frequently not prepared to handle them. For some companies the TNDS systems were the first centrally developed BIS product to be installed. Interfacing with Bell Laboratories for reporting troubles, transmitting diagnostic data, and receiving corrections represented new concepts and procedures. Of even greater difficulty, each operating company had its own local rules for documentation and computer job operations. Conventions for computer job termination, job completion, and error detection, recovery, and restart varied. In some cases local procedures required extensive manual verifications to ensure proper program job execution. From these early dissimilarities, a number of activities emerged. Working with the central development organizations and the operating telephone companies, the AT&T Information Systems organization established basic rules. And within the TNDS project, a very important activity was the establishment of "run control" standards for the TNDS/EQ and TNDS/TK systems. These controls were designed to detect error conditions as they occur, inform operations personnel of their existence, and inhibit further processing until the problem can be corrected. Once the problem is corrected, these controls permit simple restart with minimum reprocessing. Run control has been implemented in all TNDS/EQ systems and has been effective in minimizing the overhead associated with operating centrally developed software in remote batch production environments. The remainder of Section III provides details on the interface of run control with the operating environment.

3.2 Run control

The attributes of the TNDS run control standard are described in this section.

3.2.1 Load module execution sequence control

To carry out a specific program function, program modules (known as load modules) that are delivered by Bell Laboratories to the processing sites are grouped into larger entities, called jobs. The execution sequence of jobs and of the load modules within jobs usually follows a predefined arrangement. Communications between programs within a job and between separate jobs are carried out by use of intermediate files stored on tape or disk. If through operational error, a defined job sequence is violated, run control detects and inhibits further processing until the proper sequence is resumed. Before run control was used, the jobs would have continued either to completion or until an error was detected. Because of the delay in detecting the trouble, location of its source and subsequent recovery was often difficult.

3.2.2 Load module termination conventions

When a load module termination occurs, either because it has completed its task or because of an abnormal condition, it is necessary to identify the reason for its termination. This information is needed both internally by other load modules executing in the same job and externally by the computer operations staff. The run control standard specifies use of a mechanism called the condition code to indicate normal load module termination and an "abort end" or abend code mechanism to indicate abnormal load module termination. The operations staff are thereby notified of abnormal terminations in a consistent way and job processing is stopped whenever a load module terminates abnormally. Similar to the previous attribute, early detection of a trouble condition is therefore improved.

3.2.3 Restart aids

A third aspect of run control ensures that restarts after a failure are handled as simply as possible. To help achieve this, run control:

1. Automatically selects the restart load module among the set of load modules within a job and alerts operations staff in the event of an error.

2. Provides a standardized interface to the operating system "checkpoint/restart" facility, which under certain operating conditions restarts the job at an intermediate point in the processing.

3. Automatically removes those intermediate files that must be recreated as part of the restart.

4. Automatically restores the contents of update files to their pre-execution contents before beginning the restart.

Figure 2 illustrates these concepts by showing a hypothetical job sequence and listing the items that must be performed to complete a restart.

3.2.4 Logging system processing activities

Run control also specifies standards for logging information about the processing activities of the system. This information is a summary of processing activities and has been very helpful later in diagnosing problems. The following information is logged for this purpose:

1. Indication of every module executed, recorded by date, time, module name, and unique version identification.

2. Indication of when the execution of the load module was completed, and whether such completion was normal or abnormal.

3. Identification of every file processed, recorded by its name and unique version identification. In addition, for sequential files, the number of records processed is recorded.

4. Recording of any anomalous conditions encountered by the program.

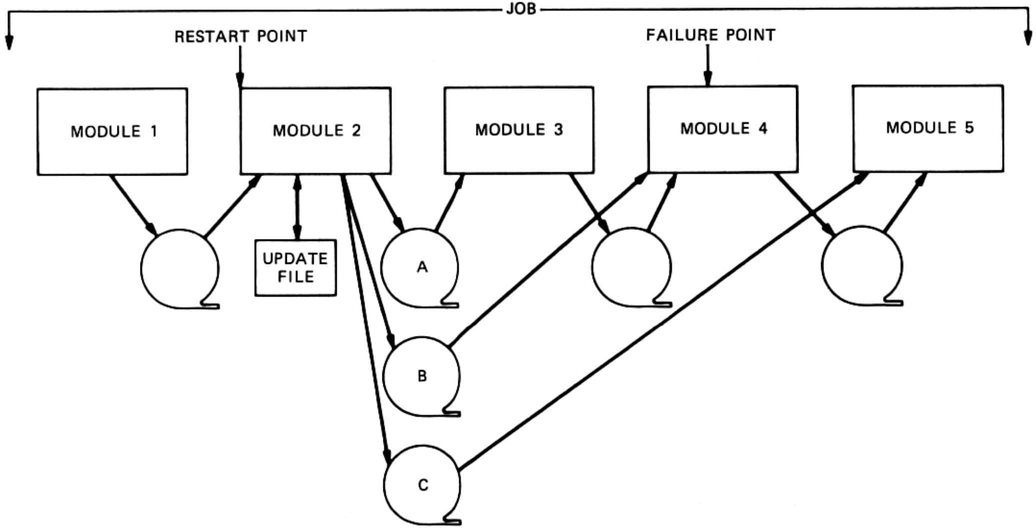
5. Recording of the Central Processing Unit (CPU) time and wall clock time needed to execute the load module.

6. Changes in value of any run control information.

The log can be readily transmitted back to the central development facility for detailed analysis.

3.2.5 Automatic verification of file versions and sequential file record counts

The fifth attribute of run control has improved the integrity of the



STATUS: SUCCESSFUL PROCESSING THROUGH MODULE 4, WHICH IS ABNORMALLY TERMINATED.

TO RESTART A JOB AFTER FAILURE:
1. DETERMINE RESTART POINT, SAY MODULE 2.
2. RESTORE UPDATE FILE TO PRERUN CONTENTS.
3. REMOVE INTERMEDIATE OUTPUT FILES (A, B, C) BEFORE INITIATING RESTART.
4. SPECIFY JOB RESUMPTION AT THE BEGINNING OF MODULE 2.

Fig. 2—Restart after failure.

file system. During a production cycle of TNDS/EQ, permanent files are modified to reflect current information. In addition to retaining the updated file, the operating system also retains the version of the file as it existed before the cycle so that a rerun can be performed if needed. The operating system provides a mechanism for using a family of files [called Generation Data Group (GDG)] to maintain information across multiple production cycles. The input to a program would be generation zero (0) and the output would be generation plus one (+1). At the completion of execution, the system records the new output as (0) and the original input as (−1). Thus generations are "rolled" from run to run. Figure 3 illustrates the use of GDGs.

Unfortunately, referencing a specific member of a GDG family requires manual intervention and can lead to accessing an incorrect member of the family. Run control detects such situations, alerts the OTC computer operations staff, and halts further processing pending correction of the problem.

Another file problem that can occur, particularly with the processing of sequential files, is loss of some of the records. A common way for this to occur is through inadvertent deletion of one of the middle reels of a multireel tape file. Run control detects this condition by maintaining knowledge of the number of records placed on the file at creation time versus the number actually processed when the file was accessed. Prior to run control, this verification was often done manually by computer operations personnel matching file counts.

3.3 Run control example

The following example illustrates how the parts of run control work together to ensure correct processing. Consider a job consisting of four load modules (LM1, LM2, LM3, and LM4). Each module creates one intermediate output file that is passed to the next load module in

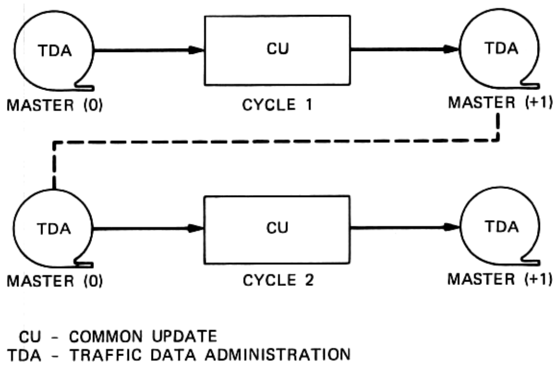


Fig. 3—Use of generation data groups.

sequence. Table III provides the contents of the run control external file. This file indicates that the current production cycle has been stopped, with only LM1 having completed execution. In this state, it is possible that LM2 might have been started, but was stopped because of an error such as a program problem, a hardware difficulty, or an incorrect tape mount.

Given the control information in Table III, when the job is executed again (the problem having been corrected), the following events occur as part of the run control functions:

1. Run control restart module executes.
   (a) A log entry is made showing run control module name, date, time, and version identification (id).
   (b) The name of the program to be executed next (LM2) is obtained from the run control external file.
   (c) The intermediate work file associated with LM2 (F2) is erased. This information is obtained from the table entitled "Association of Load Modules to Files Created" (see Table IV).
   (d) A condition code is issued to cause module LM1 to be skipped by Job Control Language (JCL) processing.
   (e) A log entry is made showing the run control module name, date, time, CPU time, and normal completion.
2. Load Module LM1 is skipped by the operating system JCL processor.
3. Load Module LM2 executes.
   (a) A log entry is made showing module LM2 name, date, time, and version id.
   (b) Run control verifies that it is proper for LM2 to execute by comparing its name (LM2) as recorded internally within the module with the contents of the run control external file.
   (c) LM2 opens file F1, extracts header information, and compares

Table III—Contents of run control external file after abnormal termination of a job

Module	Execution State	File	Date Last Executed	Time Last Executed	No. of Records
LM1	Executed	F1	8-4-81	13:00	1000
LM2	Not executed	F2	8-3-81	09:30	2500
LM3	Not executed	F3	8-3-81	09:45	5000
LM4	Not executed	F4	8-3-81	09:50	200

Table IV—Association of load modules to files created

Module	LM1	LM2	LM3	LM4
Files Created	F1	F2	F3	F4

it against the audit values in the run control external file recorded when file F1 was created. Let's assume that an earlier version of F1 was mounted and LM2 reads from the file header "8-3-81 9:00." This does not match the time and date logged in the run control external file "8-4-81 13:00." The error condition, file name, and audit values are logged.

(d) Execution of LM2 is terminated by an abend.

4. The operating system JCL processor flushes the job.

This scenario illustrates run control processing under an error condition. If file F1 had not failed its audit, then processing would have continued with modules LM3 and LM4 and actions similar to those recorded for LM2 (without the abend) would have occurred.

Program recovery prior to implementation of the controls typified by this example was often difficult, requiring extensive manual intervention. Run control software has provided dual benefits to the TNDS user community: first, it has reduced the level of direct involvement needed for centralized maintenance personnel to provide operational support; and second, it has made available better diagnostic information for handling problem situations to both the OTC operations personnel and to the central maintenance personnel.

IV. COMMON UPDATE (CU)

4.1 Introduction

Section IV describes the details of the data processing functions carried out by TNDS/EQ components. Because Common Update (CU) provides record base support to a number of the other components, it is described first. It is an example of the integration that has been achieved within the TNDS/EQ and TNDS/TK systems. In particular, CU fully supports the TNDS/EQ systems TDAS and LBS, as well as the Report Distributor facility, and provides partial support of No. 5 Crossbar Central Office Equipment Reports (5XB COER) and ICAN. It also fully supports the TNDS trunking systems TSS and TFS.

Since CU supports both switching equipment and trunking support centers, it has been subdivided into independent subsystems, CU/Equipment and CU/Trunking. This division allows an operating company to have the flexibility of providing multiple installations of TNDS Equipment (TNDS/EQ) systems, including CU/EQ, while maintaining a centralized TNDS Trunking (TNDS/TK) system, including CU/TK. The remainder of this section will focus on CU/EQ, hereafter referred to simply as CU. The article on trunking systems (Ref. 6) discusses CU/TK.

To support the Network Administration personnel who maintain the record base, CU:

1. Validates input record base transactions according to predetermined parameters and ranges, and performs intrarecord and interrecord validations.

2. Outputs activity or addenda reports after each CU cycle to provide updated record base status.

3. Provides messages and reports on transactions found in error.

4. Responds to input requests for full record base listings of a given report type.

CU executes as a series of jobs or runs. It is designed to run on an "as needed" basis, i.e., whenever transactions are created to add, change, or delete information in the record base. In most operating companies, CU executes on a weekly basis with all other dependent TNDS systems running subsequent to its successful completion.

The following four support run groupings are associated with the mainline runs of CU:

1. New TNDS/EQ installation—One-time runs needed during an initial installation of the system at a new site.

2. Release conversion—One-time runs to convert databases resulting from implementation of new TNDS/EQ features in a major new release.

3. Switching machine installations or conversions—Runs to provide CU record base support of changes in the switching or data collection environment. These include automatic database generation and conversions associated with new switching machine generics.

4. Recovery and restart—Runs that allow the database files to be restored to their original content in the event of an abnormal termination.

4.2 System inputs

CU activity is driven by batch input transactions. Each transaction is identified by a unique three-digit number, referred to as the transaction code or type. CU supports 50 different transactions, grouped into series by the TNDS system that they support. For example, transactions in the 700 series (742, 743, 750, 751, $\cdots$) support TDAS, while the 600 series transactions (600, 601, 620, $\cdots$) support LBS. Common information, which is used by multiple systems, is stored in files called tables and their transaction series number is 100 (e.g., 100, 105, 180, 181, $\cdots$).

To accommodate the current CU batch inputs many of the operating companies have developed their own front-end input processors. These systems typically are minicomputers or intelligent terminal configurations that provide screen formats, perform on-line intrarecord validations, and allow geographical dispersion of input sources.

As a future CU replacement, the On-Line Records and Reporting

System (ORRS) discussed in Section 1.4 is being developed. It will have all the benefits of the current front-end systems, as well as immediate interrecord validations, on-line record base inquiry, and on-line report viewing.

4.3 System outputs

4.3.1 General

There are two categories of outputs from CU, record base files and reports. The files support the other TNDS systems, which rely on CU for their record base. The reports support the network administrator's job of maintaining the record base so that it accurately reflects the physical telephone network.

4.3.2 Record base files

User data are entered into CU to establish an up-to-date description of the network. Information relevant to a particular processing system is isolated into one or more record base files. For example, the information that describes the data collection environment needed by the TDAS system is stored in the Traffic Data Administration master file, while the information that describes a switching entity for load balance purposes is stored in the LBS master file.

As updates are introduced to the system in the form of add, change, and delete transactions, the current version of the master file is updated to produce a new version of the file. Generation Data Groups (GDGs) are used to help simplify the updating of these files and the maintenance of adequate backups, as described in Section III.

4.3.3 Reports

CU provides a number of reports to help maintain the record base. The two main categories are error reports and reference listings.

Error reports are issued whenever CU determines that an input transaction is invalid. In addition to these reports of specific errors, other reports provide record base status and error statistics. A number of error conditions do not result in complete rejection of the transaction. Rather, to reduce subsequent data entry, the data are posted in the record base, and an error indicator is turned on. Record base status reports inform the user after each run of all records that remain invalid and require correction. Operating telephone company managers receive error statistics by transaction type and origination, which enables them to analyze error rates and spot training deficiencies. Errors that predominate in a particular geographic region of the company can thereby be recognized and corrected with additional training. Error statistics also enable managers to recognize deficiencies in system documentation.

Reference listings provide a readout of the record base. These reports take the form of addenda and full listings. Depending on the particular report, they can be generated in various formats. For example, the Data Collection Environment report can be sorted and printed three different ways, depending on the needs of the particular user.

As we described in Section IX, CU reports are directed to the report distributor for printing or microfiching and distribution.

4.4 CU processing flow

The processing flow of CU is conceptually quite simple. First, inputs are sorted by transaction type. Invalid transaction numbers are directed to an error file. The sorted transactions are then validated, starting with the 100 series transactions.

Following basic intrarecord validations, the transactions are segregated by series and directed to files for subsequent processing by individual master file update modules.

These modules update the following master files:

- Circuit group master and translation files (CU/TK module)
- Load balance master files
- Traffic unit master file (CU/TK module)
- Traffic data administration and traffic measurement requests master files
- Common access tables.

Errors detected during the update process are directed to error files. Activity and demand requests result in master file records being directed to report files. After all the files have been updated, report and error records are sorted and formatted into standard output reports.

The final step in the CU mainline process is to generate backup tapes for CU table files. These tables are all random access files updated in place in response to their corresponding input transactions. Since destruction or loss of any of these tables would be very detrimental to the entire TNDS downstream operation, the files are automatically copied during the main cycle of the system.

V. TRAFFIC DATA ADMINISTRATION SYSTEM

5.1 Introduction

As shown in Fig. 1, the Traffic Data Administration System (TDAS) is the first of the TNDS/EQ systems to process traffic data. The data are received from upstream collection systems and can be processed immediately, but typically they are processed in batches in weekly cycles. The traffic data shown in Fig. 1 are input to TDAS from a number of different sources. Among the "Other Forms of Data Collection" are vendor-supplied data collection systems and keypunched data collected in a camera/register environment. In response to specific requests for data from the users of various downstream systems, TDAS formats the data as required by those systems. As we described previously, TDAS acts as a centralized warehouse and distribution facility for traffic data.

The primary objectives of TDAS are to:

1. Accept traffic data from any standard data collection source.
2. Edit, validate, and adjust data.
3. Identify the origin of data, i.e., what the measurements represent.
4. Summarize and, if necessary, store the data, to satisfy weekly requests.
5. Transform the data into standard formats that are acceptable to the downstream systems.
6. Provide traffic data reports that primarily satisfy special study requests.

Although Network Administration is the major direct user of TNDS-collected traffic data, there are many other users. Each has different needs and, therefore, may require different subsets of the data in different physical forms and formats. The output reports of TDAS range from machine-readable files for mechanized interfaces to paper and microfiche reports.

Within the total data provisioning process, TDAS is positioned to control the data flow from the collection sources to the end users.

The remainder of this section describes how TDAS functions and its outputs, inputs, and processing flow.

5.2 TDAS outputs

5.2.1 General

TDAS supplies data to a large audience of traffic data users. Some of these are supported by computer systems with which TDAS interfaces via magnetic tape files, while others use reports created directly by TDAS. Thus, the major outputs of the system are interface files and reports.

5.2.2 Interface files

A standard interface file has been defined for each distinct system with which TDAS interfaces. This definition contains file characteristics and record formats. Associated with the record format is a detailed definition of every field within the record and its data attributes. File characteristics include definitions such as file format and block size. As user requests are processed, TDAS directs the appropriate data to the corresponding physical file, which is typically a tape but can be any IBM-compatible secondary storage device. At the completion of a weekly TDAS run, the interface files are transported to their respective data processing sites for further processing by the downstream systems. TDAS has the continuing requirement to ensure that the output formats remain stable as future changes occur in the data collection process. Note that because each class of end user has a distinct and different interest in the data, the formats serving these end users are different. For example, the format of the Trunk Servicing System (TSS) interface file is different from the Load Balance System (LBS) interface file.

5.2.3 Reports

TDAS provides data summary reports for users of traffic data not served by a downstream data analysis system. For these outputs, TDAS can perform some formatting and processing functions on the data to assist the user. These include editing, validating, adjusting for improper sample (or scan) rate, summarizing, and identifying with the appropriate Equipment Measurement Code (EMC) or Trunk Group Serial Number (TGSN) designation.

In addition to reports oriented to end users, TDAS also produces a series of reports to aid in the data provisioning and tracking process. These include:

1. Data edit error reports, tailored to each data collection source,
2. Data log reports, which relate data requests to data availability, and
3. Input data tape processing reports, which account for the many input tapes processed by TDAS in a given week.

All of these reports are written to a report file for input to the Report Distributor (Section IX) for final distribution and printing (or microfiching).

5.3 TDAS inputs

5.3.1 Record base

To achieve its objectives, TDAS relies on CU for maintenance of its record base. The TDAS record base consists of two primary inputs—a definition of the data collection environment, called the Traffic Data Administration (TDA) master file, and a list of all user requests, called the Traffic Measurement Request (TMR) master file. Using these two basic inputs, together with traffic data itself, TDAS can treat the data provisioning job as a basic order inventory problem. Orders, or TMRs, for data are compared with the inventory of traffic data. Based on prescribed rules and the use of the TDA master file for identification purposes, the orders are filled by TDAS.

5.3.2 Traffic data

Concurrent with the evolution of the switching network has been the evolution of data collection hardware. As the means of collecting data have changed, so has the format of the data itself. For example, data that originates from the old camera/register environment, which still serves some electromechanical switches, are in the format of keypunched card input. Another old form of data collection is 1 *ESS* switch data in the form of paper tape. All other modes of data collection are in machine-readable magnetic tape form but in many different formats. *ESS* switch data differ in format from electromechanical data; 4 *ESS* switch data are different from all other *ESS* switch data. Even the same switch can have different data formats based on the software version of the switching machine. One of the important functions performed by TDAS is to mask these format differences and present the data to the end user as though they were collected in a common manner.

Based on its format, the traffic data are read into the appropriate TDAS data entry module for editing, sorting, and converting them into a common format for further processing. The following input data formats are currently defined by TDAS:

1. Keypunch data in support of camera/register data collection.
2. Paper tape data in support of 1 *ESS* switch paper tape.
3. ASCII-encoded data in support of older EADAS machines; the Peripheral Bus Computer (PBC), which collects 4A toll machine data; and various outside vendor collection machines.
4. 4 *ESS* switch formatted data.
5. Binary-coded data in support of current EADAS machines and various outside vendor collection machines.

The last format, the Binary Interface, is a flexible data format intended to standardize the data coming from current stored program control systems. This interface has been documented as a General Trade *Technical Advisory* and has been distributed by AT&T through the United States Independent Telephone Association (USITA). TDAS will process data collected by any data collection system meeting the interface.

5.3.3 System options

To satisfy the variability in data collection environments and other factors that make each operating telephone company unique, TDAS accepts a number of user options, which are stored in its parameter file. One such option is the expiration date offset value for each mode of data collection. It indicates the maximum number of days from time of collection that the system will wait for all data to be received for a given type of Data Collection Unit (DCU). If all data are not received by the calculated expiration date, the TMRs will be processed with whatever data are currently available from that collection machine. This option is particularly important for camera/register collection, since each data item for this type of DCU is received as a separate and independent input. On the other hand, data collected via mechanized methods, such as EADAS, are usually received on a single input tape.

5.4 TDAS processing flow

5.4.1 General

A complete execution of TDAS, called a cycle, is performed on a weekly basis. To simplify its operation and scheduling for the computer operations personnel, TDAS is divided into a number of discrete jobs.

The mainline TDAS programs (that is, those that are run every production cycle) are divided into three functional entities: CU/TDAS File Interface Process (FIPS), Data Entry, and Data Analysis. FIPS extracts the records required from CU and formats them for efficient TDAS processing. It is a multistep run and is executed once per cycle. Data Entry serves as the front-end process of TDAS. There are five Data Entry modules (hereafter called editors), one for each of the data entry formats, as listed in Section 5.3.2. Their frequency of execution depends on the data collection environment and computer scheduling of the individual OTC. Some OTCs run daily editors, while others batch the daily data tapes into a weekly editor run. TDAS allows both or any combination in between. The editors prepare the traffic data for the Data Analysis programs, which are executed once each production cycle. This overall TDAS processing flow is shown in Fig. 4.

The primary responsibility of TDAS is to process traffic data, and that it does—in excess of 50 million data items per week in an average size OTC. It is this magnitude of data that makes TDAS the longest running system in TNDS/EQ. Run time, therefore, has been a continuing area of concern.

One of the techniques that were introduced to improve processing times is data screening. Soon after TDAS was introduced it was recognized that much of the data collected was not needed by any end user. Of the 50 million data items received by TDAS, only about 25 percent are sent downstream—the rest are excess! The varying data requirements of the different end users may result in upstream data collection for almost 24 hours of every day, even though only a small portion is needed for the entire period. Figure 5 shows in a typical example the individual needs of the downstream systems that TDAS serves and the resulting collection of unrequested data. Because of the needs of one downstream system, it becomes necessary to collect the full increment of 1000 data items for 24 hours. The function of data

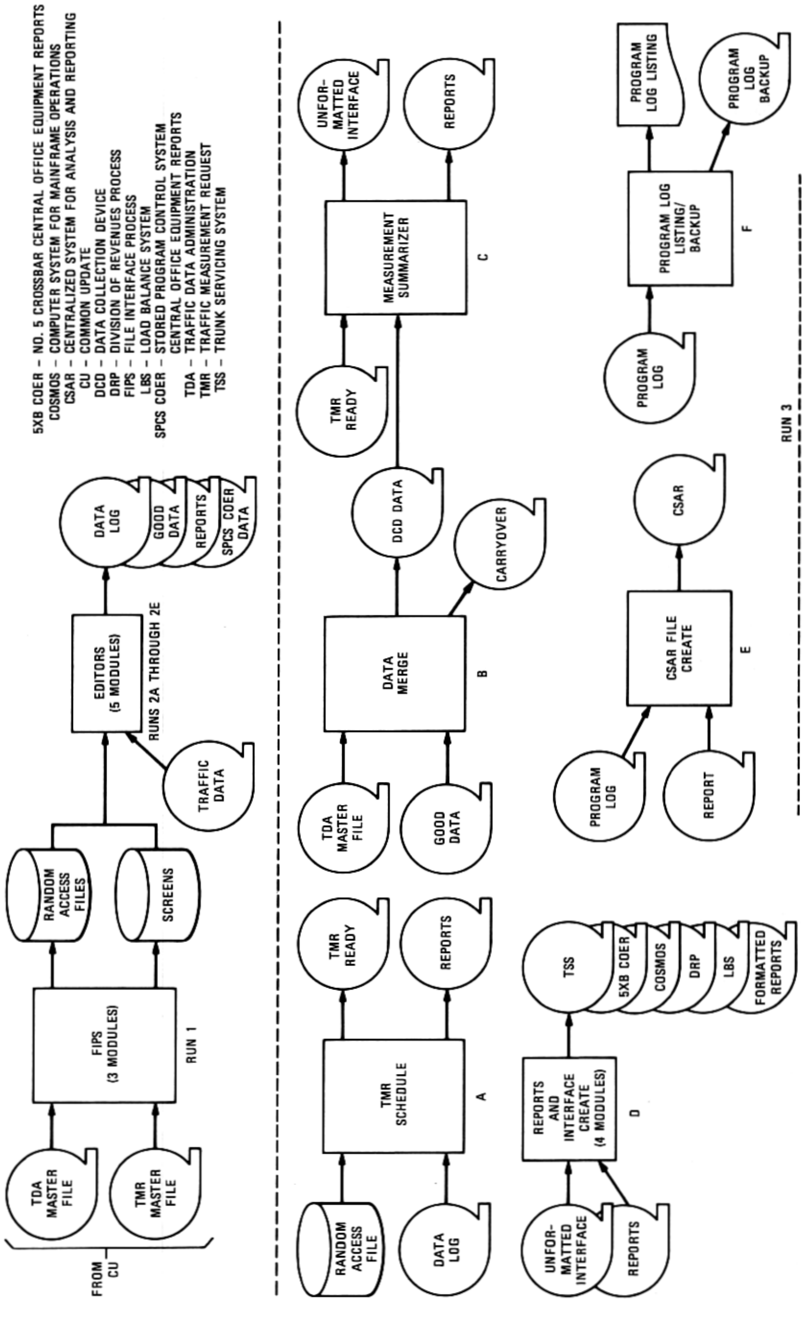


Fig. 4—TDAS processing flow.

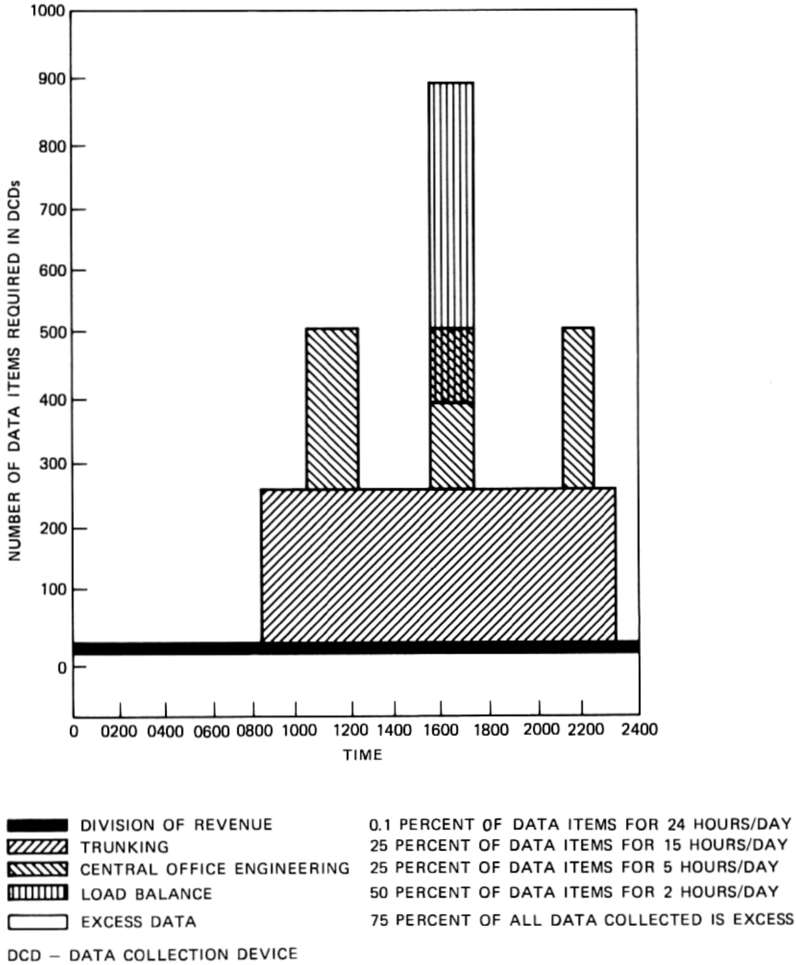


Fig. 5—Example of excess data.

screening, then, is to discard excess data at the earliest possible point in the process (the editors). It should be noted, however, that the function of screening itself incurs an expense, and for that reason it has been implemented only in those editors that process large volumes of data. The camera/register data editor, for example, does not perform automatic screening because much of the screening is done concurrently with the manual keypunching activities.

The remainder of the discussion on TDAS will focus on each of the runs that encompasses the mainline system. A number of additional support runs that perform peripheral functions, like file recovery, will not be discussed here.

5.4.2 *File Interface Process*

To perform its processing, TDAS requires certain information from the Common Update TDA and TMR master files to be accessed randomly. FIPS, then, is the first run in a TDAS cycle and is responsible for converting the TDAS master files into various random access files for efficient processing. Some files use the Virtual Sequential Access Method (VSAM), where the file is created sequentially based on a defined key and the access method routines retrieve data records randomly using indexes and pointers. Other files use the Basic Direct Access Method (BDAM), where the file is created and randomly accessed using a calculated key.

FIPS creates a two-level screen with which the editors screen the data. The programs first relate TMRs to Data Collection Units (DCUs) on a day and time basis to indicate for a given day and time whether any requests exist for data from the DCU. This first screening level is accomplished by creating a VSAM file keyed by DCU and date. The second level of screening is an interrogation of a BDAM bit map file consisting of bits for each data item, called Data Collection Device (DCD) within the data collection unit. Each bit indicates whether the data item (DCD) has been requested (1) or not requested (0). The VSAM record for a DCU/date contains calculated keys that point to maps of 1000 DCDs in the BDAM file for particular times of the day. Figure 6 shows this screen creation process.

FIPS is run following the weekly CU cycle so that the latest record base changes can be captured for the current TDAS cycle.

5.4.3 *Data entry*

After FIPS has completed its run successfully, the editors as listed in Section 5.3.2 begin execution. Each of those five editors is run separately. They can be executed in any sequence but not concurrently. An editor is responsible for detecting errors, and screening, sorting, and reformatting the acceptable data into a common output format. The editor also maintains a log of all input data. To account for lost data and the status of individual input data tapes, each editor produces a Data Errors report and a Tape Processing report. In addition, if data are lost because the collection machine goes out of service, the cause of that loss is indicated on the data tape when the collection machine returns to service. This indication is displayed on the Tape Processing report by TDAS and the same information is also forwarded to the Centralized System for Analysis and Reporting (CSAR)[2] for its total system data tracking responsibility.

The editors that process data from Stored Program Controlled Systems also strip off a portion of that data and write it to a file for transmission to the centrally located SPCS COER[1] system. The

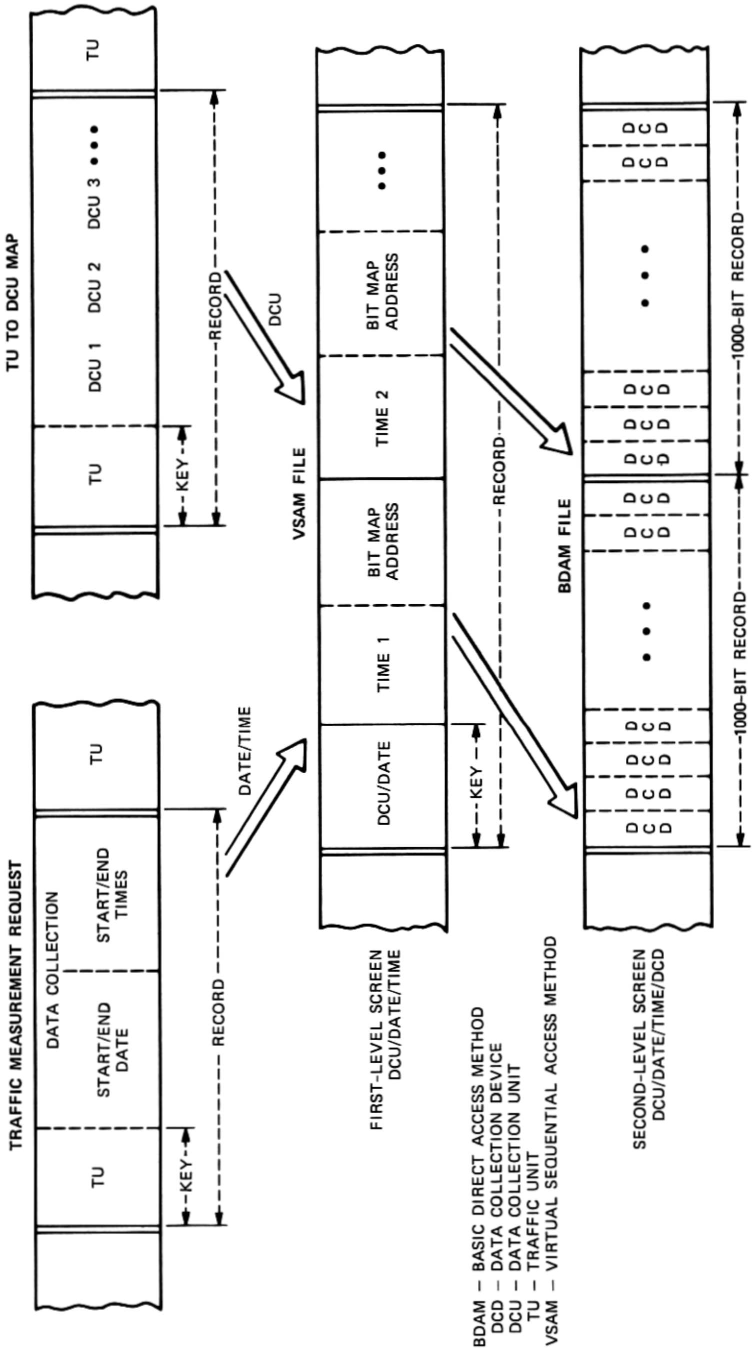


Fig. 6—Data screen creation.

TDAS-to-SPCS COER interface is the same as the collection-machine-to-TDAS interface. As a result data directed to SPCS COER are output by the editors (after being screened). Some OTCs that input daily tapes to the editors transmit the daily outputs to SPCS COER for overnight processing.

A problem inherent in the current collection-machine-to-TDAS interface is the mechanics of tape transport. In a typical OTC, 30 to 50 tapes are transported between these systems weekly. The administration and transportation costs as well as loss of tapes incurred under this arrangement have stimulated investigation into a direct collection-machine-to-TDAS data link. A future release of these systems will include the capability to transmit the data over either a dedicated or dial-up link.

5.4.4 Data analysis

After submitting all the desired data to the TDAS editors, the next and final run is Data Analysis. It is a multistep process that performs most of the work of the system.

The first step, TMR Scheduling, determines which TMRs are ready for processing in the current TDAS cycle. It compares the list of TMRs to the available data, and then based on rules and options governing calculation of expiration dates, releases whatever data are available.

The Data Merge module then merges the data with identification information from the TDA master file. The data that remains (for example, data that was input to TDAS for a partial week and does not fully satisfy the TMRs on file), along with the unexpired TMRs, are held for processing in a subsequent cycle.

With TMRs now ready for processing, the Measurement Summarizer module verifies that the collection equipment was operating correctly at the time the data were gathered. As an example, usage data are expected to be collected at a sampling rate of 36 scans per hour. If the rate falls outside acceptable limits, between 32 and 37 for electromechanical switches and exactly 36 for stored program control switches, it is marked invalid and not passed downstream. Otherwise the data are considered acceptable. If the user requests that the data be adjusted, they are then adjusted to the hourly (36 scan) equivalent. The data are then summarized into intervals as requested on the TMR and written to the appropriate interface and report files. Throughout this process statistics are compiled on invalid data and missing data.

The interface files are then sorted and formatted by the Interface Create modules for delivery to the downstream systems. Next, the Reports Create modules format the various TDAS-generated reports, and the CSAR Create module formats the CSAR interface file. Finally, the run ends with system cleanup functions.

VI. NO. 5 CROSSBAR CENTRAL OFFICE EQUIPMENT REPORTS SYSTEM (5XB COER)

6.1 Introduction

The No. 5XB COER system was one of the earliest TNDS component systems developed. It was designed to assist the network engineers in forecasting future equipment requirements in 5XB central offices. 5XB COER summarizes, over a one-year period, traffic usage data for each individual 5XB equipment component. This interval of time is referred to as the engineering year. Maintenance of this full-year history is one of the distinguishing features of 5XB COER, as compared with other systems, such as EADAS, which provide quick looks at the traffic data over short intervals of time (e.g., weekly).

The summarized data are organized into a report, referred to as the Machine Load and Service Summary (MLSS). This report is the key output of the system. The year's worth of data summarized in this report, combined with past years' data and other forecast information, enable the engineer to determine future equipment requirements.

The following sections will discuss the MLSS reports as they apply to the 5XB engineering function, and will discuss the responsibilities of Network Administration in ensuring the quality of those reports. Finally, a description of the software implementation is also provided.

6.2 MLSS reports in support of engineering

"Load" and "Service" in the title describe the content of the MLSS report: load, referring to the traffic carried by a particular component; and service, the measure of performance associated with that particular load. The engineer determines what equipment will be needed for the probable level of service to meet a projected load.

Data are summarized in High Day (HD), Ten-High-Day (10HD), and Average Busy Season (ABS) results for a Time-Consistent Busy Hour (TCBH). TCBH implies the single hour (e.g., 9 a.m. to 10 a.m.) in which the particular equipment component (e.g., originating registers for *Touch-Tone** telephone) consistently experiences its heaviest traffic load. Within this busy hour, HD, 10HD, and ABS can best be described with the help of Figs. 7 and 8. Assume in these figures that dots represent traffic data values of load on which a component is to be engineered. Figure 7 plots these values for each day (typically only business days) over the 12 months of the engineering year. As expected,

* Trademark of AT&T.

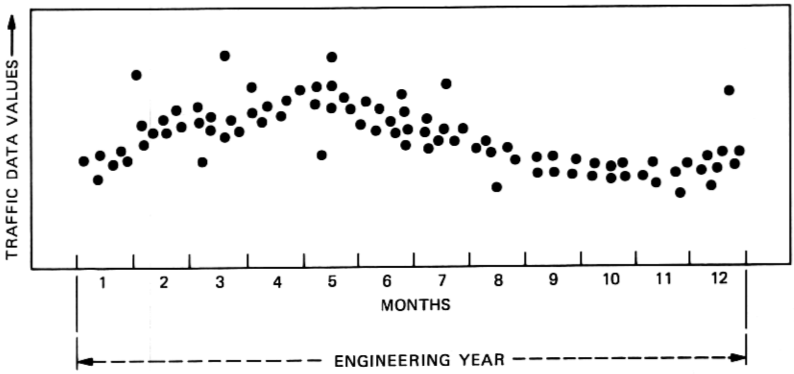


Fig. 7—Typical traffic profile.

some days have higher than normal values, and there are extended periods where the general average is higher than the rest of the year. Figure 8 shows this data summarized in terms of monthly averages for each of the twelve months, and also shows the selection of the ten highest days and three busiest months. The average of the ten highest days forms the 10HD value, and the three busiest months the ABS value.

The common format of the MLSS report used for all 5XB components, shown in Fig. 9, incorporates all the engineering requirements. Header information identifies details such as office, component, and time span. The columns (left to right) list the date, key column value* (used to engineer the component) and support columns (significant calculations such as percent overflow/blocking, holding time, percent occupancy/capacity, along with the actual values used in their calculations). The Ratio to Busy Season (RA/BS) and Fitted Ratio (FIT RAT) enable the engineer to determine if a particular reported high-day key column value is a yearly recurring event. High days typically not expected to recur (e.g., heavy telephone traffic due to a severe storm) should be excluded from the engineering base. RA/BS is the ratio of the key column value to the ABS value. FIT RAT is a statistical quantity relating this key column data point to the total year's values, assuming a gamma distribution. The RA/BS value should be very close in value to the corresponding FIT RAT value if the data point is in fact a legitimate recurring high day. The rows display the highest day (HD), the next nine highest days, the 10HD average, the next 15

* Typically, this quantity is a usage measurement normalized with respect to an office parameter such as main stations (MS) or trunks (TRK), depending on which is applicable.

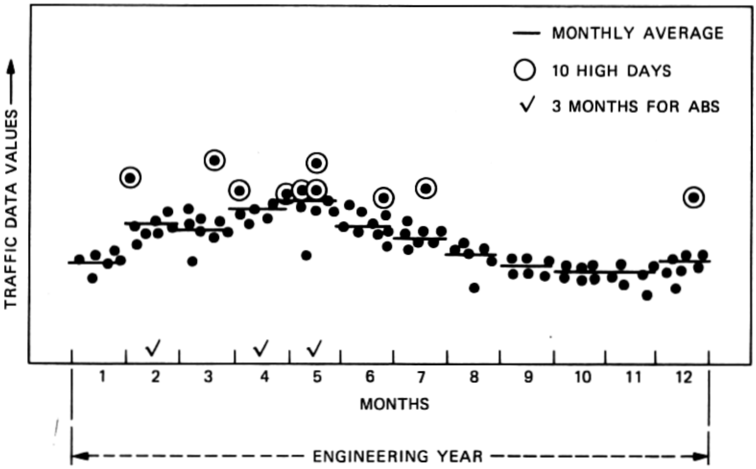


Fig. 8—Summary of traffic data.

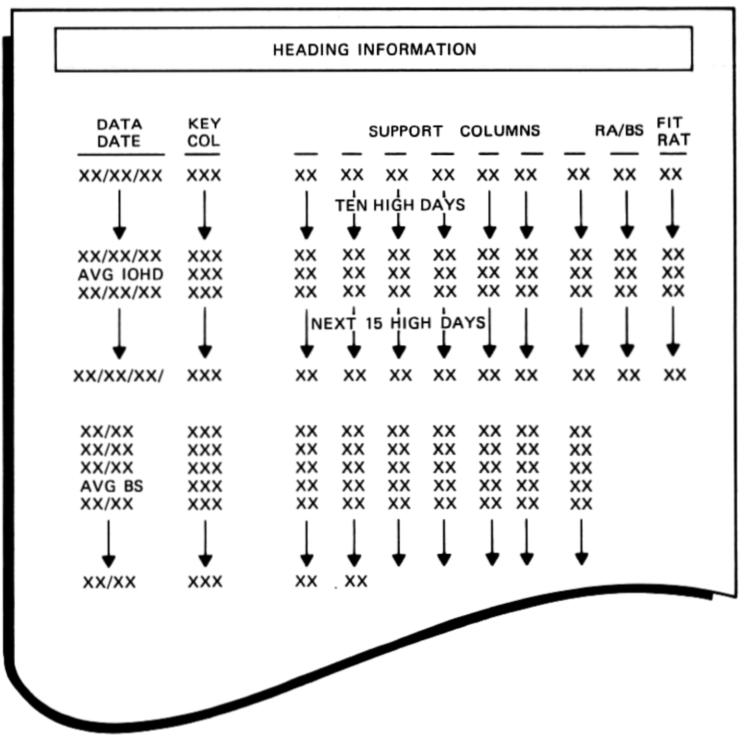


Fig. 9—MLSS report format.

highest days (displayed for their potential use as one or more 10HD replacements), the three highest monthly averages, the ABS calculation, and the monthly averages for the remaining nine months.

As traffic data are received by the system, typically in weekly batches, the MLSS report is automatically developed for each component and hour requested.* Among the components are line link frames, markers, originating registers, outgoing senders, incoming registers, trunks, and transverters.

6.3 Network administrative support

The network administrator is responsible for supplying a complete, accurate, and timely set of MLSS reports to the engineering staff. This job function fits in with the overall responsibility for monitoring the daily performance of the central office to offer the best quality of service possible with the currently installed equipment. To perform this function, equipment utilization information is gathered by coordinating the collection, report generation, and analysis of the traffic data.

To achieve the prime system objective of producing usable MLSS reports, the No. 5XB COER system supplies extensive aids to the network administrator. Among these are:

1. Busy hour determination
2. Data validation
3. Data surveillance
4. Data management.

Each will be described in the following sections.

6.3.1 Busy hour determination

Integrated into the 5XB COER system is the ability to perform a Busy Hour Determination (BHD) study. A BHD study identifies the busiest hour(s) for each engineerable component. Those hours will then be the ones reported in the next year's MLSS. Any particular component may have many busy hours contending for the heaviest load. Busy hour contention normally results from heterogeneous traffic (e.g., business versus residential). The selection of the correct hour(s) is critical because data collection and processing for the coming engineering year will be based on that decision. Selection of the wrong MLSS hour(s) can result in invalid engineering decisions.

A BHD study is conducted at least once a year, usually just prior to entering the busier part of the year. The study period typically spans

* The system will process up to five hours per component. In general, only the busy hour is studied, but one or two side hours may also be studied if uncertainty exists concerning the busy hour.

two weeks. For this study period, up to twenty-four hours per day of data can be collected in half-hour increments. The half-hour totals are summed by TDAS into hourly totals, ending either on the hour or the half hour or both. This results in up to 48 traffic data totals per day for each component measured.

Once all the study data have been collected and processed, a BHD report is produced for each component, as shown in Fig. 10. The key and support columns in this report parallel the MLSS report.

The system-selected busiest hour (based on key column) is identified with a "BH" to the left of the hour. Two additional columns, RA/BH and DAYS = BH/#, are computed to help the user understand and interpret the results, as well as to assess the degree of confidence in selecting the busy hour. RA/BH is the ratio of the individual hour to the selected busy hour. This column reveals hours that are close in load to the selected busy hour. The next column displays the actual number of days in which that hour was the busy hour (DAYS = BH), compared to the number of days that had valid data (#).

6.3.2 Data validation

In order for the summaries in the MLSS reports to help the engineers make equipment forecasts, the basic measurement data must

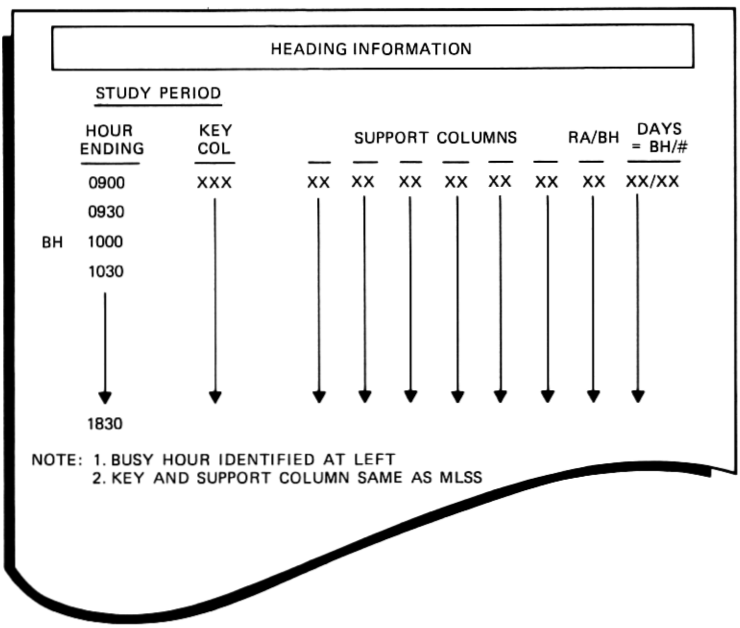


Fig. 10—Busy hour determination report format.

be a true representation of what occurred physically in the central office. An extensive set of data validation tests and associated failure reports are therefore provided.

The tests can be classified into four major categories, and they are usually performed in the following order:

1. Sanity checks
2. Peg count balance tests
3. Holding time tests
4. Historical tests.

The order of performance is based on the relative "strength" of the test. For example, sanity check failures carry a 100-percent confidence factor that the tested data are in fact incorrect. At the other end, failures in the historical tests should be investigated further before the data are declared invalid.

6.3.2.1 Sanity checks. These checks are simple and detect impossible conditions. Two examples of test failure conditions for originating registers (OR) and incoming registers (IR) are:

1. OR total usage > 36 ccs × no. of ORs in group
2. IR maintenance usage > total usage.*

6.3.2.2 Peg count balance test. This type of test defines certain relationships that must exist among several peg count (PC) measurements. For example, the Dial Tone Marker (DTM) peg count (which is scored on each call origination) should be equal to the sum of those available peg counts that indicate call disposition. In this case there are three dispositions:

1. Call successfully seized an originating register, causing the OR peg count to score.
2. Call encountered an "All OR Busy" (AORB) condition, causing the AORB peg count to score.
3. Call was not able to find an idle path through the switch, thereby causing a "Dial Tone Marker Failure to Match" peg count (DTM FM PC) to score.

The form of this test would be:

$$\text{Lower Range} \leq \frac{DTM\ PC}{OR\ PC + AORB + DTM\ FM\ PC} < \text{Upper Range.}$$

Because of the complexity of the 5XB equipment interrelationships, these types of tests often involve a large number of data items. Exact balances generally cannot be achieved because of the lack of complete measurements and the time skew for recording the various peg counts. Therefore, a range is defined around the theoretical value of 1.0.

* Total Usage means traffic plus maintenance usage.

Normally the PC balance test is performed on a component group. However, the precision of the test frequently can be refined by applying diagnostic tests to each individual in a failing group.

6.3.2.3 Holding time tests. Based on the specific equipment design, the average Holding Time (HT) of a component, e.g., Outgoing Senders (OSs), should be within a certain range of theoretical values. The range is usually broad to account for variable central office characteristics. For any given central office, the user can override and redefine the range test to strengthen the validation for that office.

6.3.2.4 Historical reliability test. The Historical Reliability test is used for data that on a statistical basis fall within a range based on past history. An example of this is the office calling load per main station, CCS/MS. The value of CCS/MS will vary according to telephone calling fluctuations but usually within a historically determined range. For these cases, tests must be designed based on past data by using a range tracking procedure.

There are many statistical techniques used for this type of validation. The method implemented by 5XB COER has produced good empirical results, was simple to implement, and required little storage. Three parameters are stored for this test: an estimated value based on past history, a variance, and a residual. The residual helps to adjust the acceptable range as the data value changes. This is particularly important in minimizing false alarms as busy seasons are encountered. The valid range for the next data point is the estimated value plus or minus one standard deviation. Based upon a comparison of the residual to the standard deviation, additional adjustments are applied to the estimated value.

6.3.3 Data surveillance

The ability to detect subtle patterns in equipment performance can mean the difference between the correct diagnosis of an equipment malfunction and the incorrect conclusion that an office is underprovisioned. Real-time surveillance is provided by systems such as EADAS, which produce exception reports based on exceeded thresholds. To perform a comprehensive analysis of total office performance, component information for the entire office must be available for the same hour. 5XB COER produces an optional report package consisting of detailed component data, validation failures, and raw measurement listing for the entire office for one or two individual hours from a week's worth of data. Network administrators select the hour(s) to be reported, and the criteria on which the system will select "the highest hour(s)" from those received to process. One of four criteria may be selected for each hour to be reported: Terminating Peg Count, Originating Peg Count, Originating plus Terminating Peg Count, Originating plus Terminating Usage. The selected reports are produced each processing cycle.

Another data surveillance feature of 5XB COER supports the annual MLSS product. As the system receives and processes the MLSS data, normally weekly, it produces output reports titled Weekly Machine Administrative Reports (WMARs) and Monthly Machine Administrative Reports (MMARs) for each component and hour requested. WMARs are optional reports. The format of these reports parallels the associated MLSS reports, except that it displays daily data for each day of the week or month rather than 10HD and ABS data. This permits a review of the daily results to ensure their appropriateness for later inclusion in the MLSS results.

6.3.4 Data management

To permit user judgment and knowledge to be reflected in the results, several capabilities can alter the contents of the MLSS report. These capabilities are called data management and include the ability to change an input measurement, change a day in the high day list, change a month in the average busy season list, or change a validation failure flag. Any changes made are identified on the MLSS so the engineer receiving the report is aware of the changes and can question their reason. The changes are made by input transactions that direct the system to modify the history for the current engineering year. At the end of the engineering year, the system produces the completed MLSS report, which reflects the entire year's data along with all data management changes. After this report is produced, the system retains the history for up to two additional months to allow for further changes.

6.4 No. 5XB COER software

6.4.1 Overview

No. 5XB COER software consists of two main procedures and four support procedures. The four support procedures perform:

1. File initialization—Establishes the record base for new installations of the system.

2. File conversion—For each new software release that affects the record base, this procedure converts from the old to the new.

3. Parameter file maintenance—The parameter file contains the information that controls the processing order of the system. This procedure can modify that information.

4. Program log list—As described in Section 3.2.4, the program log is a continuous record of system execution activities. This procedure produces a report of that information and is used primarily as a debugging aid.

The two main procedures, Control File Update and Process Traffic Data, will be described in the following sections. An overview of the processing steps and interactions of these two procedures is shown in Fig. 11.

There are two primary files in 5XB COER. The Control File contains all the user-supplied control information and office parameters. It serves a function for 5XB COER that is analogous to that provided by the CU files for TDAS and other systems. The History File contains all the processing results being accumulated over the engineering year for MLSS reporting. The data management transactions are depicted in Fig. 11 from a logical point of view as going directly to the history file; in reality the transactions pass through the Control File.

6.4.2 Control file update

The Control File Update process incorporates numerous input transaction checks to ensure database accuracy. There are two levels of checks: syntactic checks that ensure correct data formats and values, and semantic checks that perform further edits to ensure the records are logically consistent. Errors result in one of two actions: either the transaction is rejected outright, or the affected office is placed in an "error" state. The error state prevents further processing of traffic data for the office until the error is corrected. Transactions may be effective immediately or they may become effective in the future, depending on the date supplied by the user. Since this is a file update process, it is designed to run as often as necessary prior to a single execution of the Process Traffic Data run.

6.4.3 Process traffic data

Process Traffic Data is the main batch procedure of the 5XB COER system, generally executed weekly. The major functions of this process are:

1. Data Edit—Screen out all but the valid data requested for processing
2. Validate—Assess each measurement and assign a degree of confidence tag
3. Calculate—Formulate all required numerical results
4. History
    (a) Mesh new results with historical base
    (b) Perform special BHD and data management processing (if applicable)
    (c) Assemble output results file for reporting
5. Reporting—Format the output results for input to the Report Distributor for printing.

These functions are described in the following paragraphs.

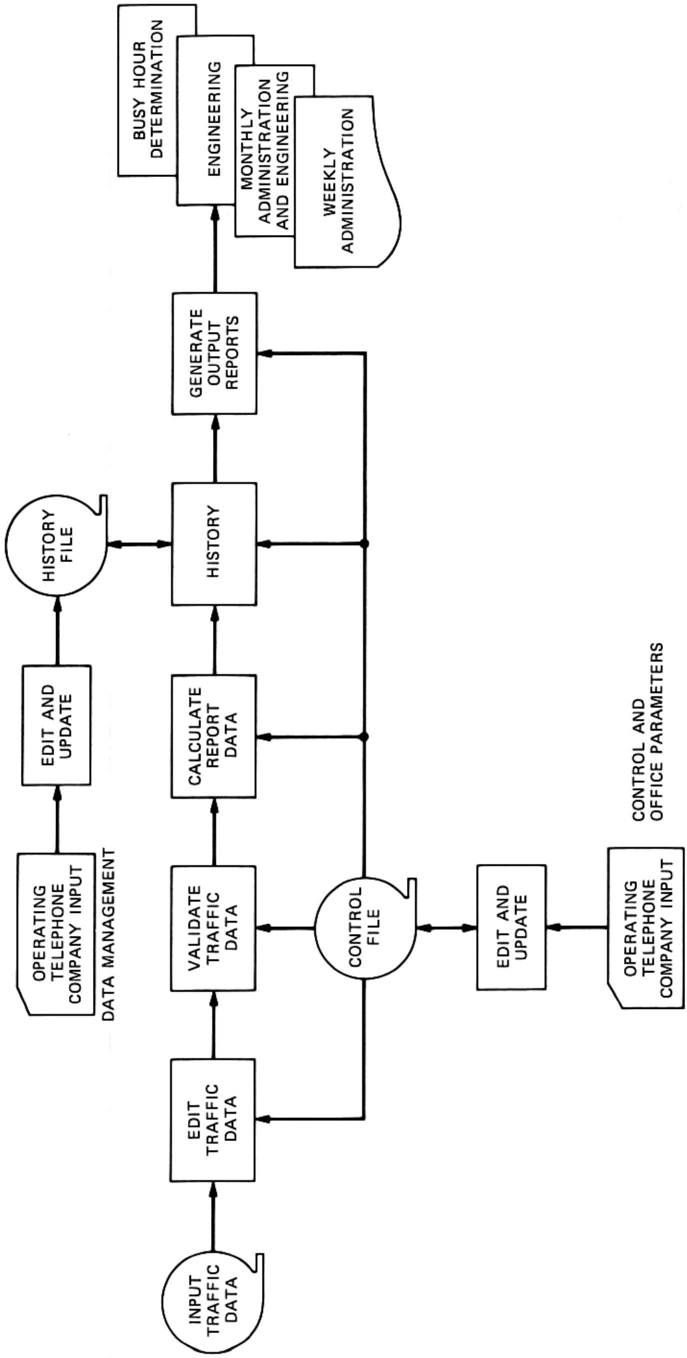


Fig. 11—Processing overview of 5XB COER.

6.4.3.1 Data edit and validate. The Edit function performs syntactical checks on new traffic data received from TDAS, places data in a Save Data File for any office indicated in the control file as being in "error", and combines new traffic data with any saved data for further processing. All input edit errors are detailed in an error report that includes a statistical log reflecting the current year's activity.

The Validation function makes a comprehensive assessment of the quality of the data by employing the extensive set of tests described in Section 6.3.2. All validation test failures are reported, identifying the data as well as test parameters involved.

6.4.3.2 Calculate. The Calculation function is the next major task. The validation tags are propagated into the calculations so that a calculation result receives the lowest tag of any of its input data items.

The calculation module has incorporated a particularly robust design that is described further because of its usefulness in other applications. The design centers on a common computation subroutine driven by tables that define the calculations to be performed. Calling routines are organized to parallel the component report structure, such that for a particular summary there is one calculation routine.

The information in the control tables is stored as a stack in symbolic postfix form. The computation subroutine evaluates the stack in two passes. The first pass checks the stack for syntactic and semantic errors and builds a work table of operand values. The second pass performs the calculation and checks for overflow and transaction errors. The control information is organized in five tables, one of which provides the calculation formulas in postfix form, the other four of which provide operand values.

Seven permissible operators have been incorporated in the calculation table: the five normal arithmetic operations of addition, subtraction, multiplication, division, and exponentiation; and two end of stack symbols, one of which indicates that a percent check should be made to ensure that the resultant value is less than 100.

This design has enabled the calculations to be easily maintained. To correct or modify any calculation, all that is required is a table update. To add a new report requires the addition of a new calling routine in which the most difficult part is the table definition.

Up through the calculation function, all the modules have operated on the current cycle's traffic data. The remaining functions, History and Report, integrate that data with previously processed data and report the results to the user.

6.4.3.3 History. The History Update module performs three major tasks. First, it merges the current traffic data with the stored traffic data in the history file. To do this, the current daily data records are sequentially added to the appropriate section of the file; the high day list is reorganized with respect to the new data with new days inserted and old days deleted; and once sufficient daily records exist, the monthly average is computed (or recomputed if it already exists). Also, if a BHD study is active and new data are received, the merging of the new hourly results by component must be completed. Second, any user data management transactions are applied to the high day and monthly average records. Third data appropriate for the WMAR, MMAR, or MLSS reports are released.

6.4.3.4 Reports. The last function executed in the process traffic data procedure is Report Formation. This module formats and merges all the outputs from the previous modules into a file for input to the Report Distributor System (Section IX). There are currently more than fifty basic report formats and over 200 variations within these formats. A multilevel table-driven design provides a generalized module that is simple to modify, maintain, and test.

VII. INDIVIDUAL CIRCUIT ANALYSIS SYSTEM

7.1 Introduction

Maintaining wiring and assignment records for traffic data collection is a complicated task. It is quite common to sample more than 10,000 points in an electromechanical switching machine. These sampled points are combined to provide accumulated counts in typically a thousand registers or data collection devices (DCDs) used for engineering and administering the office. This accumulation of sampled points into DCDs greatly simplifies the engineering and administration functions by reducing the amount of data to be analyzed. However, the record-keeping and physical wiring associated with this grouping function is a major source of error that can propagate into costly engineering mistakes.

To understand the Individual Circuit Analysis Program (ICAN) and how it assists in this record-keeping process, it is necessary to understand the functions of EADAS/ICUR. This subject is covered in detail in Ref. 4. However, Fig. 12 presents a brief overview of EADAS/ICUR.

With EADAS, traffic data counts occurring in the central office are encoded and sent via a data link to the collection machine. In basic EADAS (i.e., without the ICUR option) the usage data are collected at the output of the Traffic Usage Recorder (TUR), where the measurements have already been grouped by wiring on the cross-connect field of the TUR frame. This grouping function allows the accumulation of related information into a single DCD (e.g., usage data collected from 10 individual trunks into one trunk group). With the ICUR option, which can be applied to electromechanical switching offices, the data are grouped via software in EADAS rather than via wiring

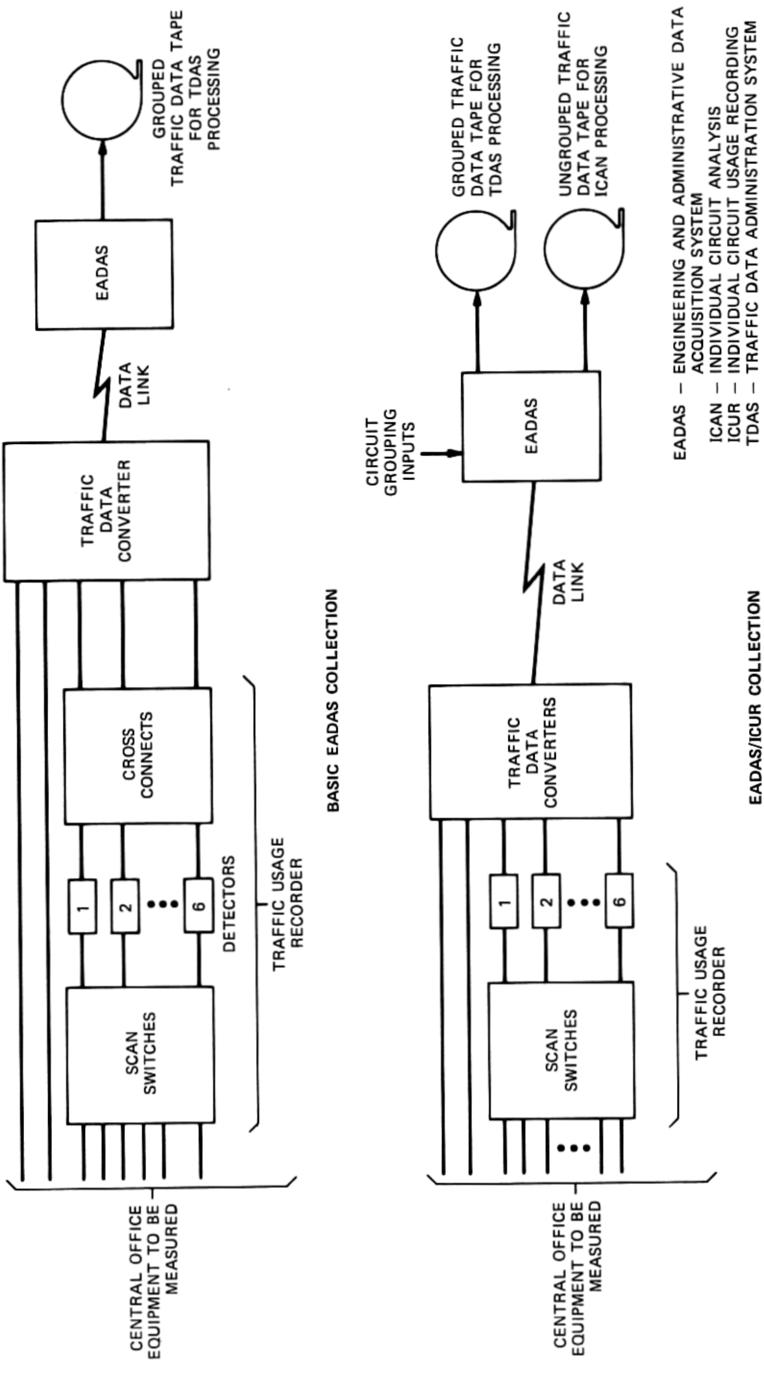


Fig. 12—Overview of EADAS/ICUR.

on the TUR cross-connect field. Software grouping simplifies the TUR administration function by replacing manual TUR cross-connect logs. In addition, the ICUR option allows network administrators to observe the usage being accumulated on each of the individual TUR inputs within a group. These ungrouped data can be analyzed to detect switching equipment problems in the central office, data collection equipment problems, and various database errors.

EADAS/ICUR produces a magnetic tape for ICAN processing. This tape contains the ungrouped usage data and additional information necessary to administer the TURs. The ICAN process is unique among TNDS/EQ components in having access to individual circuit usage information. This makes the system highly valuable in detecting central office operational problems that otherwise could go unnoticed.

This overview of ICUR operation will lend perspective to the description of the inputs, outputs, and processing flow of ICAN that follows.

7.2 ICAN inputs

ICAN's purpose is twofold: to aid in the administration of TUR frames through the use of the EADAS/ICUR circuit grouping map and to help detect faulty central office equipment items. To accomplish these functions, inputs are needed from three sources: EADAS/ICUR, the Common Update record base, and the network administrators.

EADAS/ICUR provides the ungrouped traffic data measurements, the schedules on when these data were collected, a map on how the individual measurements are grouped into DCDs, and the updates to this map since the last ICAN tape was written.

Descriptive information for each of the DCDs is obtained from the Common Update record base. Finally, the network administrators provide control information specifying the reports to be produced and the individual circuit data to be purged from the ICAN history file.

7.3 ICAN outputs

With the basic inputs described above, ICAN can produce a series of output reports to aid network administration. These reports can be produced on either a demand or exception basis. The reports are divided into the categories of record base listing reports, error reports, equipment surveillance reports, and system administration reports.

7.3.1 Record base listing reports

ICAN maintains the record base describing how individual TUR inputs are grouped into DCDs. Various views of this record base can be produced in the form of report listings tailored for a particular network administration task.

TUR administration listings provide a complete layout of the TUR inputs. Each input is identified by physical location of the central office equipment being measured, along with the associated DCD accumulating the information. To aid in the assignment of new TUR measures, a separate report can be produced itemizing those inputs that are unassigned or unequipped.

Other views of this record base list the individual DCDs and specify the associated TUR inputs that are accumulated in each DCD count. This view aids in troubleshooting various validation test failures as well as in making new DCD assignments.

A third view of the record base is organized by the equipment location being measured and identifies the associated TUR input and associated DCD. This view is primarily used in central office maintenance activities for troubleshooting reported data collection problems.

Individual TUR inputs are associated with a primary DCD. However, the scorings can also contribute to a secondary device. These secondary devices, called load grouping codes, typically accumulate total office input equipment utilization broken down by load division (see LBS, Section VIII). This secondary method of accumulation simplifies various office engineering functions by providing a single DCD representing the total usage accumulated over hundreds of other devices. Load grouping code listings show this type of DCD summarization.

In addition to the various TUR and DCD administration listings mentioned above, there are two other ICAN record base listing reports: the TUR schedule report, which identifies the ICUR data collection periods, and the Site Definition listing, which identifies all EADAS/ICUR data collection machines within the company.

7.3.2 Error reports

The ICAN error report series produces various exception reports itemizing discrepancies found in processing inputs and updating the TUR record base maintained in ICAN. The reports specify the various input transaction errors, and the discrepancies found between Common Update, EADAS/ICUR, and ICAN record bases. The record base auditing functions incorporated in ICAN correlate various parts of the TNDS record base and verify that these record bases are kept in synchronization.

7.3.3 Equipment surveillance reports

Of particular interest in the ICAN system are the three central office equipment surveillance reports. Because information is available on the basis of an individual TUR input, additional verification can be performed to assure the central office equipment is operating properly.

7.3.3.1 Abnormally Short Holding Time reports. The Abnormally Short Holding Time (ASH) report is useful in detecting a "killer trunk." A killer trunk is a circuit that reflects an abnormally short holding time because customers abandon it almost immediately to escape noise, bad transmission, or other problems. Because these trunks are quickly returned to an idle state, the switching equipment is likely to pick them more often than other trunks within a group. One faulty trunk may cause numerous failures in attempts to use the group, even though the group is well below capacity. The ASH tests detect probable killer trunks by a statistical analysis of the individual usage data.

Following is a simplified example demonstrating the working of ASH. Consider the sequence of busy/idle indications for two circuits of the same group connected to a 100-second scan TUR:

Circuit A: 111001100
Circuit B: 101001101,

where: 1 = busy condition when sampled, and 0 = idle condition when sampled. As you can see, both circuits were busy five of the nine times they were scanned by the TUR. However, Circuit A has only three transitions between busy and idle status, while Circuit B has six. If we assume normal call duration of 200 to 300 seconds, it is probable that Circuit A is carrying only two calls of at least a 200-second duration in this time period. It is also probable that Circuit B is carrying at least four distinct calls (busy indication followed by idle indication) of much shorter duration. Circuits displaying these characteristics are likely to be ASH circuits.

As noted above, many busy-to-idle transitions are indicative of a killer trunk. A busy-to-idle transition is considered a positive ASH factor. Similarly, a trunk that is detected as being busy for a number of successive TUR scans is more likely to be operating normally. A trunk appearing busy for two successive TUR scans is therefore considered a negative ASH factor.

To determine which trunks have abnormally short holding times, ICAN first weights these positive and negative ASH factors by the group occupancy rate (total CCS for the DCD divided by the number of TUR scans) and by the individual trunk occupancy for all other trunks. These weighted factors are accumulated until their sum (called the "ASH statistic") exceeds a positive or negative threshold.

If the positive threshold is reached, a message is output on the ASH report indicating that the trunk has an abnormally short holding time.

Thus, ICAN infers through a statistical algorithm that a trunk has an abnormally short holding time. Even though the ASH detection method is statistical, it is designed to be 98 percent reliable in reporting true office problems.

7.3.3.2 Unusual usage report. ICAN analyzes the usage data received and reports on TUR inputs that show usage but are unassigned or unequipped, as well as TUR inputs that are assigned but appear inactive.

Unassigned/unequipped analysis is quite simple. With each transmittal from ICUR, the unassigned/unequipped TUR inputs are analyzed for transitions between busy and idle states. If a transition has occurred, the TUR input is flagged for exception reporting. In the exception report, the amount of circuit occupancy is computed and displayed.

Inactive circuit analysis is also quite simple. If the TUR input shows no transitions between busy and idle, it is flagged for exception reporting. The state of the circuit (100 percent busy or 100 percent idle) along with the date of last activity are displayed on the exception report. Care must be taken to avoid reporting false alarms for those measurements that are typically inactive. The set of measurement types that are typically inactive are automatically identified by the unusual usage software and further analysis is bypassed.

7.3.3.3 Individual circuit usage displays. The individual circuit usage display is a valuable tool available with ICAN. At the user's prerogative, usage on selected circuits can be accumulated and displayed to help analyze problems too subtle to be detected by the ASH algorithms. These displays are also used for tracking specific troubles indicated on central office equipment. For the selected DCDs, the occupancy of each of the individual measurement components that accumulate data for the DCD are displayed in a graphical form. This enables easy comparison of measurement components with similar characteristics. For example, the occupancy over a selected time period of each of the trunks in a given trunk group can be displayed for analysis. In essence, the display report shows a view of the central office equipment operation that is similar to a time-lapse photograph.

7.3.4 System administration reports

The final set of ICAN reports is designed for administration of the ICAN System itself. This set of reports depicts activity that has occurred, and summarizes processing results for the individual ICUR input tapes on an individual DCU basis. The majority of these reports are intended for use by network administration personnel. Additional reports list computer resource usage and record base activity. The latter would be needed in the event of a system recovery.

7.4 Processing flow

7.4.1 General

The ICAN System consists of a single mainline program and several support utilities. This single job approach simplifies OTC scheduling, operation, and control. Based on control inputs, the mainline ICAN process validates inputs, processes ICUR tapes, produces periodic and exception reports, purges unneeded data, and backs up the database.

Because of the large volume of data and the complexity of some of the analysis algorithms, the system has been designed for efficient data processing. A typical company uses ICAN to administer 300 TURs, requiring the weekly processing of over 10 million individual circuit usage measures and over 30 million individual validations.

The flow of mainstream ICAN and the various utilities are described in the following sections.

7.4.2 Mainline process

The functions performed in the mainline ICAN process are shown in Table V. A single execution consists of four phases. The first, the card input processing phase, analyzes and validates the input requests, produces various error reports, and sets up the controls for the additional phases as requested by the input transactions.

The second phase, tape processing, is the most complex of the mainline ICAN and includes:

1. Schedule processing—Analyzes the ICUR tape writing schedules, produces the associated master record base updates, and outputs any appropriate error messages.

2. Card image processing—Analyzes the Circuit Grouping Map (CGM) updates produced since the last ICUR tape was written. The updates are validated and the CGMs are updated. Any discrepancies are noted on the appropriate error reports.

3. Map processing—Compares the updated ICAN CGM with the current ICUR CGM within this function. Again, after analysis, the appropriate error messages are generated and the ICAN CGMs updated to reflect current assignments in EADAS.

4. Circuit data processing—Analyzes individual circuits for ASH and unusual usage, stores data for display reports and long-term analysis, Pending Status Flag (PSF) processes, and deletes old, unneeded data currently resident in the record base.

The reporting phase, three, formulates and produces both requested and exception reports.

In the final phase a backup tape of the record base is made so that restart and recovery steps are simplified if errors are encountered in future runs of mainline ICAN.

Within mainline ICAN processing, a checkpoint feature is imple-

Table V—Mainline ICAN processing phases

I: Card Input	II: Tape Processing				III: Reporting	IV: Backup
	Schedule Processing	Card Image Processing	Map Processing	Circuit Data Processing		
Validate input cards and report discrepancies	1. Compare schedule with existing schedule 2. Report schedules if changed 3. Determine number of tape writes on tape	1. Validate and report discrepancies 2. Update ICAN maps or discrepancies	1. Compare 1-for-1 with existing maps 2. Report discrepancies 3. Change ICAN maps	1. Store data 2. Unusual usage-inactive circuit 3. PSF change 4. Produce ASH report 5. Display report 6. Delete data if requested	Assemble and produce reports	Generate backup tape

mented so that if processing problems are encountered later in the same job, job restart will begin automatically at the point immediately after the last successfully processed ICUR tape.

7.4.3 *Support process*

To supply needed DCD descriptive information to mainline ICAN, a single support run is periodically executed to extract the DCD information from the Common Update record base, identify discrepancies between the record bases, and produce the necessary record base updates.

7.4.4 *Utilities*

A number of support utilities are available to be used when needed. These utilities create and modify various portions of the ICAN record base, provide recovery capabilities of the ICAN record base, and serve as a backup in case problems occur in the EADAS/ICUR record base.

VIII. LOAD BALANCE SYSTEM

8.1 *Introduction*

A principal Bell System goal has always been to offer the best possible service at the lowest possible cost. For the network administrator, this objective translates into making the most efficient use of existing switching facilities while maintaining an acceptable level of service to all customers. To achieve this goal, network administrators must distribute the telephone traffic load evenly across the machine's switching (or load) units. This principle is called load balance. This section will describe a tool available to help achieve that goal—the Load Balance System (LBS). LBS is a integrated component of TNDS/EQ. It relies on CU for maintaining its record base and on TDAS for supplying the weekly traffic data with which it does all its processing. Distribution and printing (or microfiching) of all LBS reports are the responsibility of the Report Distribution Facility (see Section IX).

LBS supports all 1 *ESS* switch, 2 *ESS* switch, No. 1 Crossbar (1XB), 5XB, Crossbar Tandem (XBT), and Step by Step (SXS) machines for which data are collected. Network administration functions are supported by calculating a Load Balance Index (LBI) and providing assignment and balance guides.

8.2 *LBS outputs*

8.2.1 *Index reports*

Because of the importance of load balance, an administrative index was devised in the mid-1960s to measure the effectiveness of load balance procedures. LBS mechanizes the implementation of the revised Load Balance Index Plan, as prescribed in AT&T administrative practices. Index reports are produced on a traffic unit (TU) to indicate the level of service and balance achieved by that TU. From these measures indexes are aggregated into the hierarchy of district, division, area, and company scores. On a Service Observing Month (SOM) basis, the company-level report is forwarded to AT&T.

For those offices in which the index is not calculated by LBS, typically smaller, rural SXS machines, the index can be entered into the LBS record base (via CU) for inclusion in higher-level LBI reports.

8.2.2 Data summaries

Data summaries provide detailed information on the data used in calculating the Load Balance Index. These reports require constant attention. Identified on these reports are load units that have rising or falling load trends, unusually high or low loads, or penalty points assigned from the index calculations. These are signs of either current or potential trouble spots in the switching machines and indicate not only where corrective action may be needed but where additional studies may be appropriate. These reports should be used in association with the line assignment aids (see Section 8.2.4) to achieve an increase of traffic in the more lightly loaded load units while diminishing traffic in the more heavily loaded load units. This is typically accomplished through normal disconnect and assignment activity. In drastic cases, reduction of heavy loads is accomplished through specific transfer of customer lines out of heavily loaded units into lightly loaded ones. This procedure to improve line load balance (and thus the Load Balance Index) should result both in increasing the machine's effective capacity and improving service to the customer.

8.2.3 Second Session Studies reports

Many switching entities experience a secondary daily busy hour during some part or all of the year, due to the calling habits of a community of customers with different calling patterns than those responsible for the primary busy hour. While the LBI is calculated using only primary busy hour data, it is also important to provide quality service during the secondary busy hour. To identify secondary busy hour imbalances, LBS produces Second Session Study reports. The same principles apply to use of second session data summaries as to the busy hour data summaries. The reports are similar, with the exception that penalty point and index values are not calculated for second session studies.

8.2.4 Line Assignment/Transfer reports

LBS produces two basic aids for line assignment. Line Assignment

Guides provide lists of load units, with the most lightly loaded ones listed first, thus appearing in order of preference for assigning new lines. Line Equipment Transfer Guides start at the other end of the list; the most heavily loaded units appear first and are thus in order of preference for transfer of lines. The user may request a Line Assignment Guide or Transfer Guide in any length desired. These reports assist the manual process of determining new line assignments and, if necessary, transferring current assignments to other load units. As telephone companies introduce mechanized line assignment processes, such as the Computer System For Main Frame Operations (COSMOS), manual assignments from LBS reports are discontinued. For that reason Line Assignment Guides are produced only on demand on a TU basis.

8.2.5 Nonindexed balance aids

There are some components such as trunks and service circuits that are also sensitive to load balance but are not included in the LBI calculation. To assist in achieving proper balance of those components, LBS produces balance aids that contain corrective load values.

8.3 LBS inputs

LBS requires three categories of input to perform its functions: traffic data, traffic unit characteristics, and a history base. The history base is a file internally maintained by the system itself. The other two inputs are provided by TDAS and CU, which operate with LBS in a tightly integrated fashion.

8.3.1 Traffic unit characteristics

For LBS to judge the degree of balance within a traffic unit and to provide aids in achieving or maintaining proper balance, the system requires a detailed description of certain TU characteristics. This information is provided by the user via CU. Through the input of various transactions, the user builds a Load Balance Master File, which contains a description of all TUs that LBS supports, specifying the traffic measurements to be processed for each TU. Included in such TU descriptive information is common language identification, number of main stations, destination of output reports, assignment and loading division information, theoretical capacities, and average holding time (AHT).

8.3.2 Traffic data

In addition to TU characteristics, LBS requires data that represent the traffic load being offered to the TU. The source of the data is, of course, the switch itself. However, before reaching LBS, the data are first obtained by a data collection system and passed through TDAS. Using the traffic measurement definitions and traffic measurement requests, TDAS validates, adjusts, and summarizes the data before writing them to an interface file for LBS processing. TDAS also applies an Equipment Measurement Code (EMC) label to each data item, permitting LBS calculations to be driven by internally interpreting the EMCs.

8.3.3 History files and system parameters

Since some of the computations involve averaging and trend analysis, it is necessary that history files be maintained for reference and updated on each run of LBS. Included in the history files are the previous week's loads and penalty points, average holding times, percent capacities, and total-to-sample ratios. The latter is a means of determining total usage on a load unit from a sampling of two of its links.

Certain numerical values are required in the LBI algorithms used by LBS. These include default values for average holding time and total-to-sample ratio, Quality Control Limits (QCL), designation of threshold values for "hot-spot" determination, and specifications of LBI correction values. The QCL is a measure of how far the load on a load unit is allowed to deviate, without penalty, from the average load for the loading division.

8.4 LBS processing flow

8.4.1 General

A complete execution of LBS is performed on a weekly basis following the running of TDAS. The system is divided into six discrete runs, one for mainline LBS and the other five serving as support runs. The remainder of this section will address the mainline run of the system, as shown in Fig. 13.

8.4.2 Study formation and validation

The Study Formation and Validation module receives the input traffic measurements from TDAS. It performs validation tests on the traffic data and combines it with office characteristic information obtained from the CU files for further processing. In addition, if the appropriate peg count data are available, average holding time (AHT) and total-to-sample (T/S) ratios are calculated for later use in index and corrective CCS calculations.

Validations are categorized as follows:

1. Study identification—Data are checked to ensure that only valid combinations of studies have been requested.

2. Data volume—The number of hours requested is not always the

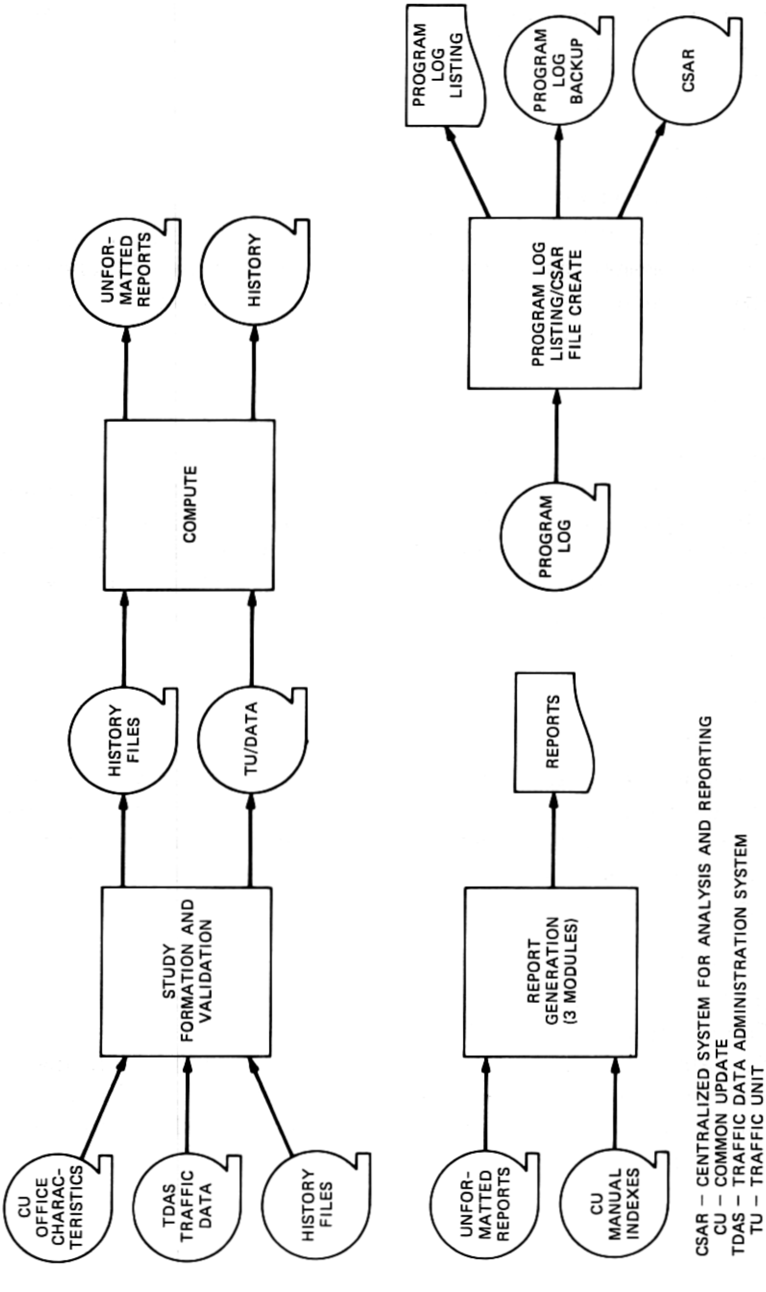


Fig. 13—Load Balance System processing flow.

number received. Acceptable ranges for the different studies have been established to allow for reasonable variation in equipment performance and to retain the statistical reliability of the study. If sufficient data are received, the study is abandoned. In addition, usage data for any hour having either zero or more than 36 CCS are rejected.

3. Data identification—Data are validated to contain acceptable EMC and measurement-type designations.

4. AHT and T/S Ratio—Calculated AHT and T/S ratios are compared to historical values. If they differ by more than a fixed percent, they can be discarded, based on the judgment of network administration personnel.

5. Study criteria—To perform an index study, more than 75 percent of the load units in a TU must have valid data and the loading divisions must be at greater than 30-percent capacity. If both conditions are not satisfied, the study is aborted.

8.4.3 LBS compute

Following the above data validations, the Compute module performs the many calculations necessary to satisfy the study requests. The first function is to react to history change requests to recompute, delete, and reinstate data from history.

Component calculations that contribute to the LBI are performed next. Using AHT and T/S ratio, the quality control limits are determined. Scores are then developed depending upon the indicated deviation of actual usage from the average usage. Load units operating at loads above predetermined thresholds are assigned a "hot-spot" penalty. The index is then calculated using weighted linear regression techniques that utilize up to 12 weeks of past data. Since load imbalances affect service more severely as the total traffic handled by the office increases, the percent load capacity is another factor used in determining the final index.

Finally, corrective CCS values are derived from linear regression estimates of what each load unit's deviation from the average will be at the time of the next study. These values are displayed on the balance guides in the appropriate order to facilitate manual corrections to imbalance.

8.4.4 Reports formations and system cleanup

The remaining modules assemble, sort, and format the various reports produced by the system, and perform cleanup functions.

IX. REPORT DISTRIBUTOR

9.1 Introduction

For the effective utilization of the TNDS downstream systems, it is essential that the reports be distributed in an efficient and timely manner to the users, most of whom are located physically remote from the computation center executing TNDS. The TNDS/EQ components alone can produce several hundred different report formats for distribution to as many as 600 distinct locations within an operating telephone company. To complicate this process further, selected reports are needed on microfiche produced by equipment manufactured by several different vendors, and other reports require multiple copies.

Early in the evolution of TNDS, a Report Distribution and Printing Facility was developed. This capability is utilized today for all the downstream TNDS systems. In addition, it is available as a standalone package and can be used by non-TNDS systems to solve their report distribution needs. The following sections describe the capabilities of the Report Distributor, its inputs, outputs, and processing flow.

9.2 Report Distributor capabilities

The design of the report distribution process is based on two standards implemented throughout the downstream TNDS systems:

1. Every report type is uniquely identified with a five-character report identification specified in the report header.

2. Every report utilizes a four-character responsibility code indicating the specific user who is to receive the report. This responsibility code is also identified in the page header.

With these two basic standards, the report distribution process was designed to produce:

1. Microfiche—For long-term storage, or cost reduction reasons, users are able to select particular reports for output to microfiche. The Report Distributor has the capability of producing outputs that meet the specifications of several microfiche hardware vendors.

2. Multiple copies—Users are able to request multiple copies of selected reports without multiple passes through the distribution process.

3. Proprietary marking—The users can specify that the appropriate proprietary notice be printed on the individual report pages.

4. Monitoring distribution—Although the primary individual is identified with the report responsibility code, alternate distributions can be identified, usually for monitoring purposes.

5. Burst pages—Before a printout can be distributed, it must be broken down by responsibility code. To make this easier and to reduce errors, burst pages are optionally printed (three pages of output with characters printed over the fold) between reports where the responsibility code changes.

6. Remote print—Where the report destinations are physically remote from the computer center running TNDS, time and cost

savings are realized by producing report tapes according to destination and having them printed locally at the remote site. This facility allows selected reports to be routed to numerous locations.

7. Selective retrieval—To recover lost listings, or for historical checking of a report, selective retrieval can be made of an individual report or all reports for a particular responsibility code from a report tape.

8. Print restart—Following a system failure or job cancellation, the report printing program is able to restart at a point that results in minimal duplication of printout. Typically, report print programs are run with a dedicated printer, so that restart points can be communicated to operations through the system console. The frequency of the checkpoints are at the discretion of the OTC. When restarting is needed where multiple print tapes are involved, the restart does not have to spool through complete tapes that have already been printed.

9.3 Report file inputs

The report distribution facility imposes two requirements on every system providing a reporting file for distribution:

1. The report file must be organized in responsibility code/report number order.

2. Key records must be included at specific points within the report file. These records identify the number and responsibility code of the report following the key record. At a minimum, a key record is specified on each change in responsibility code and report number. In the worst case, a key record is provided for every report page. (This is necessary if the report is on microfiche and an index to the fiche is provided on a page basis.)

The report distribution process detects these key records and through the use of Common Update files determines the proper output distribution device.

9.4 Common Update inputs

Inputs through Common Update define the pattern for the distribution process and the types of proprietary messages to appear on reports. Three types of transactions are necessary to define these characteristics:

1. One transaction defines the report output device. These devices are used to break up the distribution process into parts. An individual device may be designated for transmission to a remote computational center where the reports on that device are further subdivided for distribution. A device may be designated for input to a particular type of microfiche equipment, or it may represent the set of reports to be distributed to an individual user.

2. The second type of transaction specifies which report number/responsibility code combinations are to be distributed to each of the numerous output devices.

3. The third transaction specifies by report number the proprietary message to be printed at the bottom of each report page.

The current contents of the various report distribution tables maintained by Common Update can be examined on request.

9.5 Outputs

For problem-tracking considerations, output reports are produced that summarize the information written to each of the output devices during every execution of the report distribution process. These reports identify each responsibility code/report number change, along with any further report content information contained in the key record.

9.6 Report distribution process

The report distribution process is implemented by a combination of three programs (Fig. 14): the merge program, the distributor program, and the printer program.

9.6.1 The merge program

All TNDS report generation programs produce report data sets ordered by destination (responsibility code), similar to the outputs from the distributor and printer programs. When multiple report tapes are to be processed by the distribution facility, a merging of the reports must be made. This is the function of the merge program. It is designed as a separate run because (1) its output may be input to either the distributor or printer program, and (2) to put the merge functions in the distributor program would increase the number of devices required to run it.

9.6.2 The distributor program

The distributor program must be used when there is a need for the generation of remote print tapes and/or microfiche tapes. It uses as input the TNDS keyed report tapes, a key definition file, and the report distribution table. It produces any combination of three outputs: keyed report tapes, nonkeyed report tapes, and microfiche tapes. Keyed report tapes are printed at a local or remote site and can be fed to the printer program directly or further processed at the remote site by another copy of the distributor program before being printed. Nonkeyed report tapes are printed at a local or remote site, and microfiche tapes can be processed by any of the supported microfiche systems.

Report processing by the distributor is key record driven. That is,

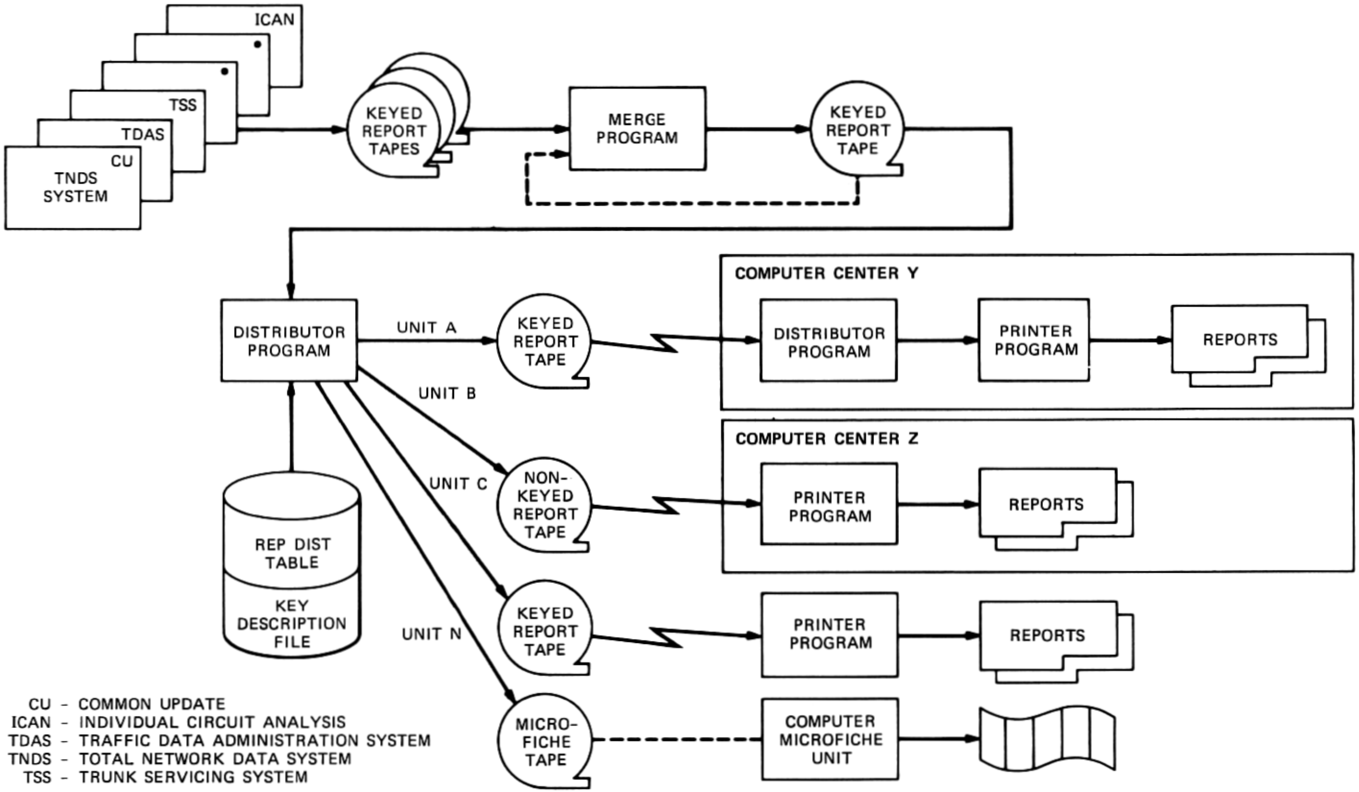


Fig. 14—Report distribution process.

when it encounters a key record on a report tape, it matches the responsibility code and report number against the selection masks in the report distribution table, identifying those report devices that are to receive the report. It then treats all report records up to the next key record as one report, spooling each record to the appropriate data set(s). If no report selection mask matches the report, the report is written to a default report device. When the report is completed, it produces its own reports summarizing the processing that has occurred.

To accommodate changes easily in the meaning of key record fields for reports, allow for addition or deletion of reports by a system, and permit use of the distribution facility by other than TNDS systems, the distributor program has no system-specific internal information. All report-dependent information (i.e., legal report numbers, algorithms for formatting key record data on microfiche, index lines, and distributor reports) is kept external to the program, in a Key Definition file constructed by the using systems.

There are four options supported in the distributor program for microfiche output, which are provided by the device definition record in the report distribution table. These options deal with blank microfiche separators, magnification factors, forms flash, and microfiche sequencing. The blank microfiche option, if used, specifies that a blank microfiche is to be generated when the responsibility code changes. This allows operations to more easily cut the microfiche film for distribution. The magnification factor parameter specifies the magnification to be used in generating the microfiche. The forms flash option enables the user to have a preprinted form superimposed over a microfiche frame. Finally, the microfiche sequencing option determines whether the sequence number as printed on the microfiche is to be reset to one when a new responsibility code is encountered. Each report number or responsibility code change causes a microfiche advance.

9.6.3 The printer program

The printer program can be used both at the OTC central computer center and at a remote print site to produce hard copy listings for distribution. It has options for burst pages and printer restart (options under control of EDP operations) and selective report retrieval and multiple copies (under control of the TNDS EDP coordinator).

X. MANAGEMENT REPORTING SYSTEM (MRS)

10.1 Introduction

In the past, the OTCs originated many requests to develop specialized reports in addition to the current fixed-format reports available through the various TNDS components. The Management Reporting System (MRS) was developed to answer this need. MRS enables both TNDS/EQ and TNDS/TK users to extract information available in various TNDS files and format this information to satisfy their local reporting needs.

Besides customized report generation, MRS offers two other major capabilities for its users. First, MRS can be used to mechanize the production of Common Update input transactions. This capability is especially useful for initializing an office database and for various office conversion activities.

Second, MRS can serve as an interface between software processes. Data may be extracted and made available to OTC-developed systems without the user having to know the actual structure of the TNDS source, nor care about future revisions to this structure.

In its implementation, MRS is an adaptation of the MARK IV™ software package of Informatics, Incorporated, Canoga Park, California. Using this package, Bell Telephone Laboratories personnel provide the necessary file definitions, interfacing software, and associated documentation required to produce reports, transactions, and system interfaces.

10.2 File definitions

More than twenty major TNDS files are defined for MRS use. The file definitions describe the record characteristics and define the structure of each segment within a record. These definitions allow a convenient interface between the TNDS products and the flexible reporting capabilities made available to the user community through MRS. BTL provides the file definitions and updates them with each major release. Because the Mark IV definitions describe the data information items independent of file structure, OTC-designed programs need not be modified as the basic files are restructured with new TNDS releases.

10.3 Mark IV interfacing software

In addition to the basic file definitions and documentation, BTL provides various MARK IV interfacing software. This software ranges from catalogued requests to restructure the TNDS files for MARK IV use to BTL written code that is integrated with MARK IV providing specialized translation operations. Examples of these specialized translation operations include:

1. Conversion of Trunk Group Serial Numbers to the Common Language Circuit Identification (CLCI).
2. Conversion of Location Machine Processing Codes to Common Language Location Identifications (CLLI).
3. Date and time format conversion routines.

10.4 Using the system

To illustrate the use of MRS, consider a request by a TNDS/TK user to identify, list, and count all trunk groups that use a particular group as an alternate leg in completing a call. Using one of the MRS reference manuals, the MARK IV definitions of the required information are identified. In this example seven distinct field definitions are used:

Field definition 1—The Trunk Group Serial Number (TGSN) of the group being investigated.

Field definitions 2 through 4—The TGSN of the first, second, and third alternate legs at the A end of the circuit group.

Field definitions 5 through 7—The TGSN of the first, second, and third alternate legs at the Z end of the circuit group.

A MRS request is now coded on standard MARK IV forms. The forms specify the comparisons to be made on the input files, the sort sequence of the data appearing on the output report, and the format and field titles of the output report.

In the above example a scan is made of the A and Z alternate legs for the particular TGSN in question (DV000189). If DV000189 appears as an alternate leg, the trunk group is listed in TGSN order with all legs identified. Finally, a count of the TGSNs using DV000189 is computed and displayed. Figure 15 shows an example of the resulting output report.

10.5 Processing flow

The generic flowchart for an MRS run is shown in Fig. 16. Standard TNDS files and MARK IV user source are submitted as input to the run. The specified fields are identified and analyzed in Step A, during

03/27/82 GROUPS USING DV000189 AS ALTERNATE LEGS PAGE 1

TGSN	TGSN AL1	TGSN AL2	TGSN ZL1	TGSN ZL2	TGSN ZL3
DV000170	DV000189	DV000173	DV000173	DV000189	
DV000183	DV000196	DV000189	DV000189	DV000196	
DV000187	DV000165	DV000189	DV000189	DV000165	
DV000186	DV000175	DV000189	DV000189	DV000175	
DV000142	DV000168	DV000189	DV000189	DV000168	
DV000143	DV000164	DV000189	DV000189	DV000164	

03/27/82 GROUPS USING DV000189 AS ALTERNATE LEGS PAGE 2

TGSN	TGSN AL1	TGSN AL2	TGSN AL3	TGSN ZL1	TGSN ZL2	TGSN ZL3

GRAND COUNT 6

Fig. 15—Example of a Management Reporting System report.

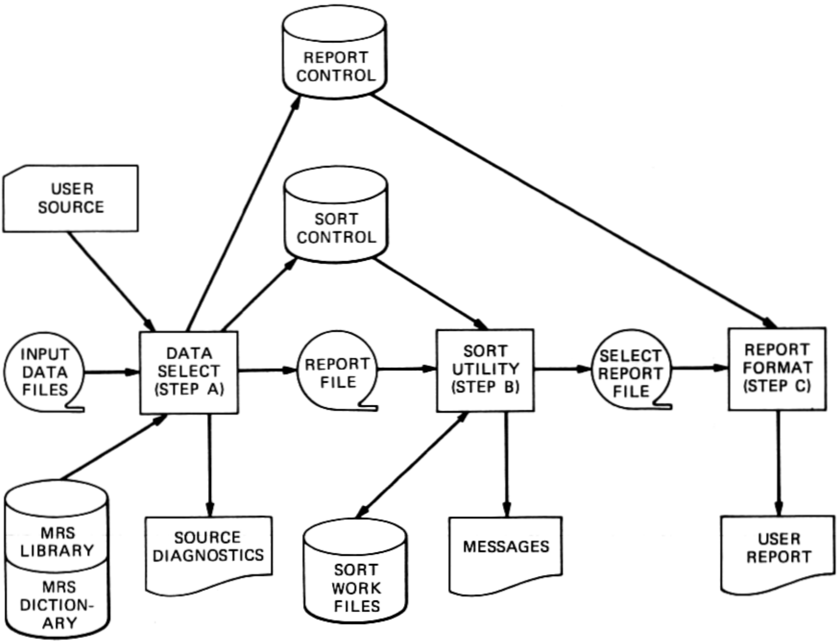


Fig. 16—Generic flowchart for the Management Reporting System.

which the basic report file is produced. Step B is a sort utility that organizes the report file into the sequence requested by the user. Step C takes the organized report file and formats the output report requested by the user. If the objective was to generate a TNDS transaction file or create a file for input to other mechanized systems, the sort output file is used directly, bypassing the third step.

XI. SUMMARY AND FUTURE DIRECTIONS

Components of TNDS/EQ have been in use since the early 1970s. These systems play a key role in the overall administration of network data. The systems are in production use by all Bell System operating telephone companies and Bell Canada. Extensive controls have been implemented to ensure smooth and reliable operation of the data processing center. Through standardization of the data collection environment, interchange of data among companies of common items has become possible. Development and user support activities are carried out by Bell Laboratories and Western Electric central developers, with enhancements to the systems provided by annual releases.

For the future, the major work for TNDS/EQ will be twofold: to maintain support of the continuously evolving central office and trunking network, and to provide more responsive service by using the on-line record base and reporting features of the On-line Records and Reporting System (ORRS). This replacement to Common Update will be developed during the period from 1983 to 1986. The modernization of TNDS/EQ will include establishing telecommunications links between collection machines and TDAS, integrating the No. 5XB COER Control File into the ORRS record base, and assimilating functions currently carried out by CSAR into ORRS. In assessing future telephone company needs to be supported by TNDS/EQ, the key will be responsiveness. The use and expected number of users of traffic data will increase, along with demands for more timely, reliable, and accurate data. The modernization program for TNDS/EQ will be essential for these new business requirements.

REFERENCES

1. R. F. Grantges, V. L. Fahrmann, T. A. Gibson, and L. M. Brown, "Total Network Data System: Central Office Equipment Reports for Stored Program Control Systems," B.S.T.J., this issue.
2. D. R. Anderson and M. J. Evans, "Total Network Data System: Performance Measurement/Trouble Location," B.S.T.J., this issue.
3. D. H. Barnes and J. J. O'Connor, "Total Network Data System: Small Office Network Data System," B.S.T.J., this issue.
4. C. J. Byrne, D. J. Gagne, J. A. Grandle, Jr., and G. H. Wedemeyer, "Total Network Data System: Data Acquisition and Near-Real-Time Surveillance," B.S.T.J., this issue.
5. M. S. Hall, Jr., J. A. Kohut, G. W. Riesz, and J. W. Steifle, "Total Network Data System: System Plan," B.S.T.J., this issue.
6. P. V. Bezdek and J. P. Collins, "Total Network Data System: Trunking Systems," B.S.T.J., this issue.
7. W. S. Hayward and J. Moreland, "Total Network Data System: Theoretical and Engineering Foundations," B.S.T.J., this issue.

AUTHORS

N. Dudley Fulton, B.S.E.E. (Electrical Engineering), 1967, Purdue University; M.S. (Computer Science), 1972, Ohio State University; Bell Laboratories, 1972—. Mr. Fulton has worked on several components of the Total Network Data System family including TDAS, SXS COER, 5XB COER, and TRFS. He is currently Supervisor of the Trunk Forecasting Software Development group. Member, Tau Beta Pi, Eta Kappa Nu, IEEE, and ACM.

James J. Galiardi, B.S.E.E. (Electrical Engineering), 1966, Illinois Institute of Technology; M.S.E.E., 1968, Ohio State University; Bell Laboratories, 1966—. From 1966 to 1970 Mr. Galiardi was located in the Columbus Laboratories and was involved in the initial hardware and software design of the 4A ETS toll switching machine. In 1970 he became involved in mechanizing various switching machine laboratory support functions in minicomputer hardware and software. From 1971 to 1973 his work responsibilities included hardware diagnostic programming for Common Channel Interoffice Signaling. In 1973 he was promoted to a development supervisor position in the Total Network Data System Project and relocated to New Jersey. From 1973 to the present Mr. Galiardi has held various supervisory positions within TNDS, ranging from system planning through the development process to system test. Member, Eta Kappa Nu, Tau Beta Pi.

Edward J. Pasternak, A.B. (magna cum laude, Engineering Sciences), 1957, Harvard University; M.S.E.E., 1962, E.E. (Professional), 1966, Columbia University; Bell Laboratories, 1962—. Mr. Pasternak was initially assigned to an Exploratory Development group which was examining stored program control methods for modernization of SXS. With the inception of the Traffic Service Position System (TSPS) project, he was assigned development responsibilities for the TSPS Audit programs. In 1966 he was promoted to Supervisor of the TSPS Audit and Recent Change Programs Group. Following cutover in 1969 of the first TSPS office in Morristown, New Jersey, Mr. Pasternak was appointed Head of the Stored Program Control (SPC) Development Department with hardware and software design responsibilities for the SPC 1A processor. In 1973 Mr. Pasternak was transferred to the Business Information Systems area of Bell Laboratories as Head of the BISCUS/FACS System Test Department. In 1974 he became Head of the Servicing and Estimating Department, with design responsibilities for several systems that were subsequently incorporated into the Total Network Data System. Mr. Pasternak currently has responsibilities for the TNDS Equipment Systems component of the Total Network Data System. Member, IEEE, ACM.

Stewart A. Schulman, B.S. (Computer Science), 1972, City College of New York; M.S. (Computer Science), 1974, Stevens Institute of Technology; Bell Laboratories, 1972—. Mr. Schulman began his career at Bell Laboratories in the System Test area of the Service and Estimating (S&E) department, which later evolved into the Total Network Data Systems—Equipment (TNDS/EQ) development organization. Following his initial assignment, he was involved in the development of various TNDS/EQ component systems. In 1979, Mr. Schulman was promoted to Supervisor, responsible for ongoing development of the Traffic Data Administration System (TDAS) and the Load Balance System (LBS). His current assignment is development Supervisor for the On-line Records and Reporting System (ORRS), an on-line replacement for an existing TNDS batch system and the vehicle for future TNDS/EQ modernization.

Herman E. Voigt, B.S.E.E., 1962, Polytechnic Institute of Brooklyn; M.S.E.E., 1964, Columbia University; Bell Laboratories, 1962—. Mr. Voigt's initial work was in exploratory development of Common Control for Step-by-Step local switching offices. From 1963 to 1973 he was involved in the development of the first Traffic Service Position System (TSPS). His prime areas of contribution were in the software supporting the operator actions call processing functions, for which he received two patents, systems test and eventually field support of all installations. Since 1973 he has been supervising the development of various operation support systems that provide administrative and engineering equipment utilization reports for local electromechanical and electronic switching offices. Member, Eta Kappa Nu, Tau Beta Pi, IEEE Computer Society.