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

Total Network Data System:

Data Acquisition and Near-Real-Time Surveillance by EADAS

By C. J. BYRNE,* D. J. GAGNE,* J. A. GRANDLE, Jr.,* and G. H. WEDEMEYER†

(Manuscript received May 3, 1982)

The Total Network Data System (TNDS) requires a facility to collect the traffic data generated by electromechanical and electronic offices. The Engineering and Administrative Data Acquisition System (EADAS) fulfills this function. EADAS systems that serve electromechanical offices employ unique self-testing circuitry to interface to central office signals. A novel buffering scheme also improves system efficiency. For electronic offices, a specialized file system has been developed, and the input data are specified in a high-level language. All of these features permit EADAS to be a cost-effective, flexible, and reliable data collection system. EADAS also provides real-time surveillance and reports for the switching offices it serves. A set of reports is predefined; however, the user may also specify reports using the Network Operations Report Generator feature (NORGEN). System capacity is specified in simple formulas which are derived and presented. With these features, EADAS provides a comprehensive facility for presenting real-time traffic data to the telephone companies.

I. INTRODUCTION

High-quality telephone service requires facilities that collect and process data on the traffic being handled by a telephone system. In

* Bell Laboratories. † AT&T.

©Copyright 1983, American Telephone & Telegraph Company. Photo reproduction for noncommercial use is permitted without payment of royalty provided that each reproduction is done without alteration and that the Journal reference and copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free by computer-based and other information-service systems without further permission. Permission to reproduce or republish any other portion of this paper must be obtained from the Editor.

particular, it needs data on the number of calls being handled by the system, the duration of each call, and the number of times a call encounters difficulties as it proceeds through the system to its destination.

Such data are needed for surveillance of the quality of service being provided. This requires, for example, measuring in real time the interval required to obtain dial tone, and informing responsible personnel when such intervals exceed acceptable limits.

Traffic data are also used to determine the proper quality and distribution of central office equipment and trunk facilities. Other uses are to provide telephone customers with special reports on the performance of their own particular facilities and grade of service, and to assist telephone company personnel in proper maintenance of equipment.

In the Bell System, traffic data are collected and processed by several different systems. Among these are the Telesciences, Inc. "Automatic Traffic Recording and Analysis Complex" (AUTRAX) System, the Conrac Corp. "Alston Traffic, Engineering and Management Information System" (ATEMIS), and the Western Electric "Engineering and Administrative Data Acquisition System" (EADAS). The operation of EADAS will be described in more detail in succeeding paragraphs to illustrate how traffic data are typically collected and processed.

II. SYSTEM CONCEPT

Figure 1 shows the overall system concept of EADAS. A centrally located minicomputer collects data from the central offices. Within certain limitations described later, these offices can be electromechanical or electronic and can be large or small, with and without toll features.

Traffic data are collected via a data-link facility between each central office served and the EADAS central unit (CU). As mentioned above, EADAS provides information to telephone company network administration personnel for near-real-time surveillance on the quality of telephone service being provided. EADAS also makes available information to assist maintenance personnel in remedying equipment problems before they can affect service. This information is in the form of reports produced by the Network Operations Report Generator (NORGEN) feature of EADAS. These reports can be in printed form or displayed on a Cathode Ray Tube (CRT). NORGEN produces a wide variety of such reports, which are made available on demand, on a previously established schedule, or automatically when an abnormal service condition (exception) develops.

NORGEN was introduced in 1977 to replace the demand for re-

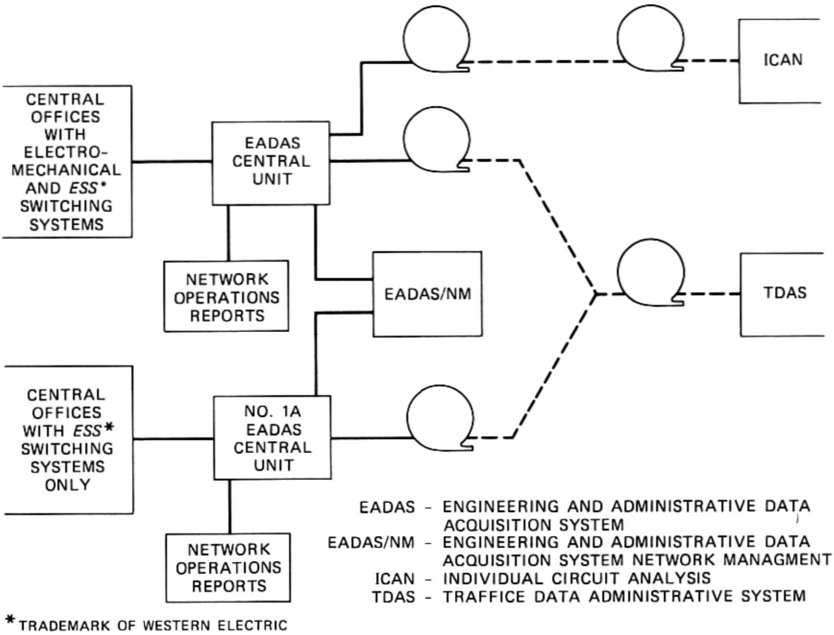

Fig. 1—The Engineering and Administrative Data Acquisition System concept.

porting features that had exceeded the calculation capacity and flexibility of the original reporting features of EADAS. NORGEN provides a calculation capacity six times larger and with greatly increased flexibility, as well as a set of standard applications programs designed to meet those near-real-time report needs required on a systemwide basis. A programmability feature allows the user to modify the standard reporting features and to provide new ones to meet local conditions.

Figure 1 shows that traffic data collected by EADAS can also be recorded at regular intervals on magnetic tapes. These tapes are physically transported to the Traffic Data Administration System (TDAS) and to the Individual Circuit Analysis (ICAN) program described in the article on TNDS equipment systems[1] in this issue. TDAS formats the data for use by other downstream software systems. These systems report on longer-term engineering functions, such as determining how much of a switching system's capacity is being used and forecasting future trunking requirements. ICAN analyzes subsets of the data collected by the Individual Circuit Usage Recording (ICUR) feature of EADAS to detect record base errors, equipment faults on individual circuits in electromechanical offices, and malfunctions in the data collection and processing functions performed by the central office and EADAS facilities.

Figure 1 also shows the data link between EADAS and the EADAS/

Network Management (EADAS/NM) System. Over this link, traffic data are sent at five-minute intervals to alert the local Network Management Center or Regional Operations Center to network problems. Signals from EADAS/NM, which activate various network controls such as trunk reroute, and signals to EADAS/NM, which indicate the status of these controls, are also transmitted over this link.

III. FIELD OF APPLICATION

There are two versions of EADAS. The original system, No. 1 EADAS,* uses a 16-bit minicomputer with 128K words (word equals two bytes) of memory. This system was designed to serve No. 1 and No. 5 Crossbar, Crossbar Tandem, and Step-by-Step type electromechanical offices, as well as 1/1A and 2/2B *ESS*† switching equipment, and Remote Switching System (RSS) offices. The No. 1 EADAS was first installed in Kansas City, Missouri, in 1973. In 1974, the ICUR feature was first installed in Miami, Florida.

A later version—the No. 1A EADAS—was designed to serve *ESS* switching equipment offices only. The No. 1A EADAS employs a faster minicomputer (three times the processor speed) with expanded memory capability (larger address range and use of cache memory). The No. 1A EADAS has approximately twice the capacity for serving *ESS* switching equipment offices, with a lower installed cost than No. 1 EADAS and the flexibility to grow with the continued expansion of Stored Program Control electronic switching in the Bell System. All No. 1A EADAS installations have the NORGEN feature. The No. 1A EADAS was first installed in New York City in 1977. The No. 1A EADAS is presently arranged to serve 1/1A, 2/2B, 3, and 5 *ESS* switching equipment offices, as well as RSS offices. In addition, coverage is provided for Northern Telecom's DMS-10‡ offices.

The No. 1 EADAS met the Network Administration needs of electromechanical offices. However, the Bell System trend toward adoption of Stored Program Control (SPC) for new and replacement offices is reducing the number of electromechanical offices in service. This, together with the advantages of No. 1A EADAS for SPC entities, has caused a leveling off in the number of No. 1 EADAS installations. On the other hand, the number of No. 1A EADAS continues to increase in proportion to the installation of Stored Program Control offices.

IV. DATA COLLECTION

The data collection process differs between the electromechanical offices and the electronic offices. Electromechanical data are collected

---

* The offical designation of the original system is simply "EADAS."

† Trademark of Western Electric.

‡ Trademark of Northern Telecom.

from several terminal types. The predominant terminal type provides real-time data, while data are also accumulated and held by other types of terminals in end offices. The accumulated data sources operate over either a dialed-up or dedicated link, resulting in three different interfaces for the electromechanical offices. On the other hand, electronic offices send only accumulated data to the EADAS, and each electronic system has a unique output format. The data collection process in electromechanical offices will be discussed first.

A number of requirements were considered in developing the electromechanical design:

1. It was advantageous to use the data collection equipment of the existing Traffic Data Recording System (TDRS) already installed in a number of central offices. The main concern here was the high cost of installing new traffic measurement equipment in electromechanical offices. This is caused by the large number of points to be monitored, which are scattered throughout the office and require many individual wiring runs.

2. Data were to be collected in real time to provide information for exception reports and the network management feature (see other articles on Network Management).

3. The system was to provide new features such as Individual Circuit Usage Recording (ICUR), described later, and office control functions for the network management feature mentioned above.

The traffic data equipment used by the then existing TDRSs, known as the Traffic Data Converter, operated as follows. The unit, located in the central office, is connected to the data collection center via a dedicated link. The occurrence of an event in the central office (i.e., generation of a call or detection of a busy on a usage scan) causes the converter to generate a data word representing the input address or connection point for the event. Thus, the function of the data collection center is to sort and sum the occurrences of these address words by address for the desired measurement period. While this function is simple in concept, it can quickly become complex to provide a capacity of 100 data channels, each capable of generating 1000 input addresses. Since all of the real-time data are available at the collection center at any instant, centralized accumulation is ideal for real-time data collection. Thus, the network management and real-time reporting functions can be readily accomplished.

4.1 Input network

The desire to provide new features and a capacity greater than that of the TDRS led to the design of a new converter for collecting data that could be used in those offices not already containing TDRS equipment. In addition to collecting peg counts, this design provided remote office control, fast scan usage data, individual input scaling (to reduce the data channel load for rapidly occurring events), concentration, and ICUR. This converter, called the EADAS Traffic Data Converter or ETDC, had a number of interesting features. Perhaps the most interesting is the input circuitry used to interface the system to the central office. This input circuit is shown in Fig. 2. The circuit has to provide a high impedance input which could monitor ground closures provided to relays (without changing operating characteristics), while absorbing inductive spikes and contact bounce associated with the relays. In addition, a periodic self-test should be provided. The input circuit provides these features as well as an input verification feature.

Meeting the high impedance requirement was a problem. Transistor-transistor logic (TTL) of the early 1970s employed 2K pull-up resistors, and a quick analysis will show that to achieve a low on the input, the maximum resistance to −48V could be no greater than 20K. This was not anywhere near the 100K to 300K desired. By power switching the TTL gate, the high input impedance can be obtained since the TTL gate is effectively removed until the input is scanned. A capacitor (C) added to the circuit to remove noise pulses serves as a charge storage point and holds the input state during the short interval when the power is switched onto the TTL gate. With this arrangement, the levels fed to the TTL gate are determined by the external voltage divider R_P and R_I.

This network is also testable. If the power switch is turned on permanently, all of the input capacitors will charge through the TTL pull-up resistor, and all of the input points should read a high state if

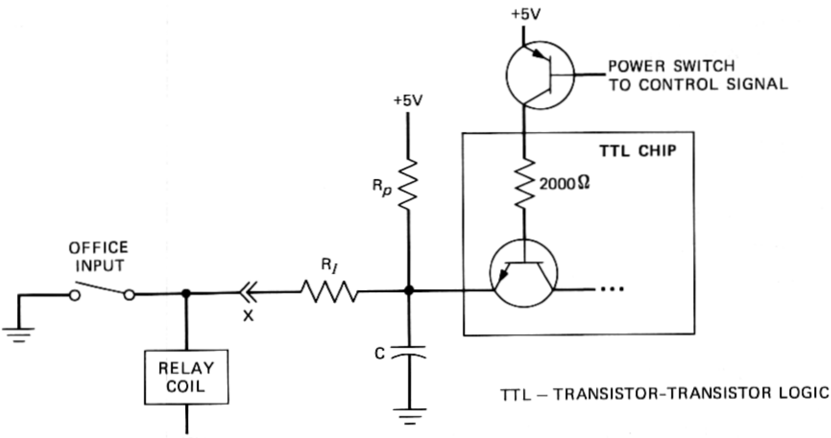

Fig. 2—Input circuitry used to interface EADAS to the central office.

a failure has not occurred. To test the low state, the +5V supply is removed from the resistor R_P. This results in all inputs being low since the voltage feeding the capacitor is either negative or zero.

Another test feature is available. Note that if a positive voltage is applied to the input at point X in Fig. 2, it is possible to introduce a high input on the TTL gate. Thus, if one takes a positive supply and pulses an input, it should register a count. This feature has two uses. First, input connections can be verified in a central office by going to the input source and pulsing it with positive voltage. Even though other inputs are active, only the positive input will count. Second, using this same technique, any crosses to other inputs will appear as additional counts or generation of additional addresses.

4.2 Electromechanical CU operation

Having described some features of the central office circuitry, let us turn to the central data collection operation. As stated earlier, this requires the ability to collect data from up to 100 links with 1000 addresses each. The simplest approach would be to use 100,000 words of memory and use the incoming address along with its link number as the address of a location to be incremented by one. Needless to say, this operation is memory intensive, and considering that the capacity of the minicomputer is 128K words, not very efficient.

A more appropriate implementation is to store the data on a bulk storage device, such as a fixed head disk, and buffer the incoming data until a sufficient amount is collected to update the disk. Once again complications arise. Disks can only be written one sector or one track at a time. For the disk selected, 2048 words represented a track. This was used to hold two incoming channels with 48 words for header. Since the rotation speed of the disk is 1800 rpm, and two rotations are required to update two channels (one read and one write), the maximum rate for updating 100 channels is

$$1800\ \frac{\text{rotations}}{\text{minute}} \times 1\ \frac{\text{channel}}{\text{rotation}} \times \frac{1\ \text{update}}{100\ \text{channels}} = 18\ \frac{\text{updates}}{\text{minute}},$$

if no other accesses are allowed. However, in actual operation the data must be read and used for processing so that only half of all disk accesses are allocated to input processing. Therefore, nine channel updates/minute are provided. Thus, the incoming data must be buffered for one ninth of a minute or 6.67 seconds. Since the arrival rate at maximum channel capacity is one address every 12.5 ms (an address is 15 bits long including start, stop, and parity and is transmitted at 1200 baud), then a buffer of size 534 is required for each channel. For 100 channels, the buffer contains 53,400 words of memory.

If one looks at actual data storage needs, one would find that at any given time nearly half of the cells in these buffers are empty. Thus, in reality only 26,700 words are needed if an algorithm can be developed to access the space. This is done using a buffering scheme that employs the triangular geometry discussed below.

First, assume that every time data are gathered into the buffer for all channels, only one channel can be updated from the buffer to the disk. Second, assume a system with five channels. Then, the buffering structure will be as shown in Figs. 3 and 4. The process is as follows. For the last channel updated (initially use channel 1), place the scanned incoming data sequentially in the vacated locations, starting with the first entry in the row assigned to that channel (channel 1 has row 1; channel 2, row 2; etc.). When the right row edge is reached, the remaining data points are placed in the column under the right edge. This operation is shown for five scans in Fig. 4, where once five scans are completed the appropriate cells for the data from the next scan are empty and the process can be repeated. Generation of addresses for using this algorithm on a digital computer is quite simple. The cells are numbered as shown in Fig. 5. Then, given:

$$S = \text{Scan or channel being processed,}$$

$$X = \text{Start address of buffer} - 1,$$

$$N = \text{Number of input sources,}$$

a. To find the first location C_1,

$$C_1 = X + S$$

b. To find the remaining locations 2 through N,

$$C_A = C_{A-1} + N - (A - 1) \quad \text{for} \quad 2 \le A \le S,$$

$$C_A = C_{A-1} + 1 \quad \text{for all other } A, \quad S < A \le N.$$

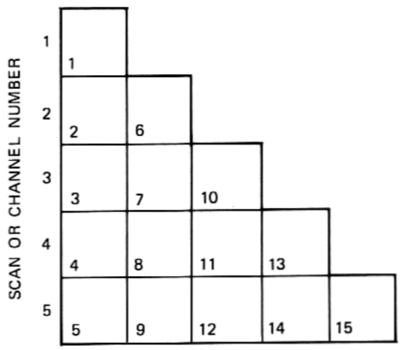

Fig. 3—Buffering scheme used to develop the algorithm that determines actual data storage needs.

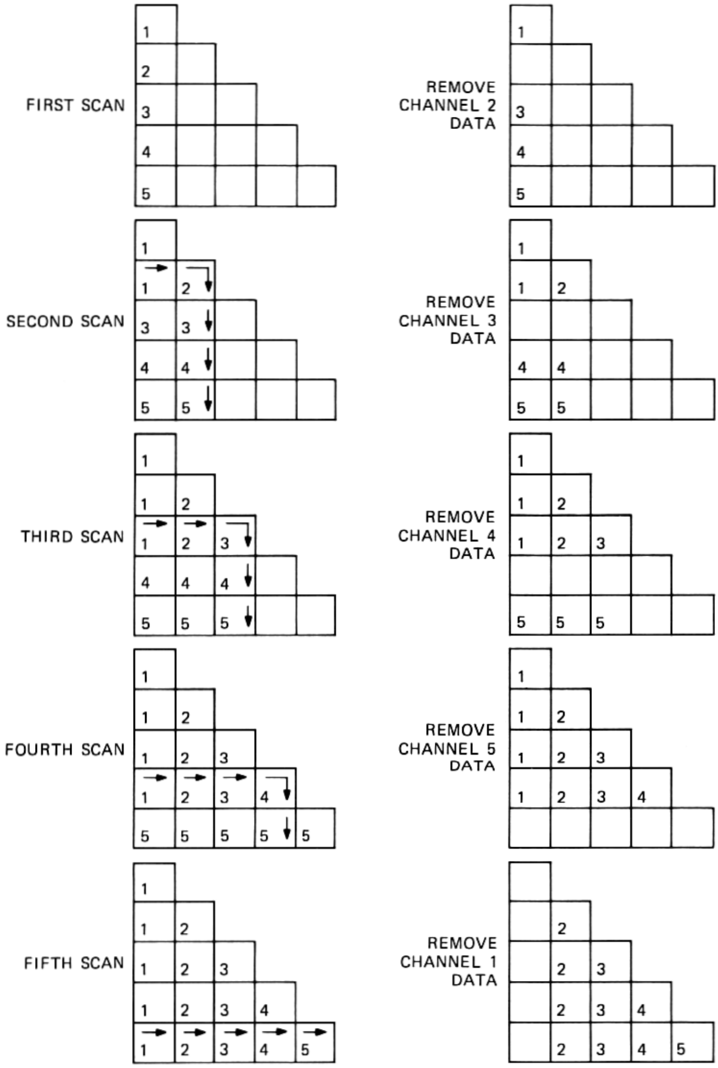

Fig. 4—Remaining data points in column under right edge of buffer structure, for system with five channels.

The above example assumed one update process for each data scan. In EADAS, scanning must occur once every 12.5 ms, while the update process takes four disk rotations for every two channels or two rotations per channel, which is equal to 66.6 ms (i.e., 66.6/6, which is the implementable rate closest to 12.5 ms). If scanning occurs every 11.1 ms, then six buffers of edge size 100 can be employed. This requires 6 $(100^2 + 100)/2$ memory locations, or 30,300. This number is slightly

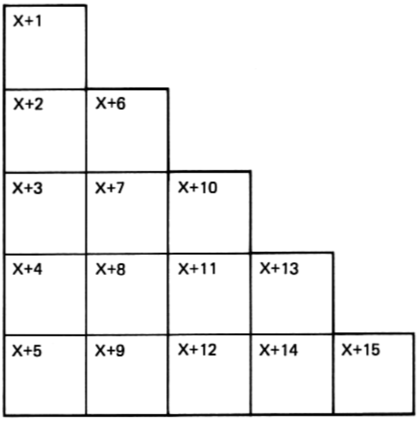

Fig. 5—Scheme for numbering buffer so that it is a contiguous string of cells.

larger than required by the 11.1-ms actual scan rate. Thus, with this algorithm, memory requirements for data collection were reduced, resulting in additional space for other functions.

In addition to regular group usage (data which are collected over all trunks in a group), the EADAS Traffic Data Converter (ETDC) includes circuitry to provide Individual Circuit Usage Recording (ICUR). These data require individual transmission of each of 3600 inputs on up to four 4A Traffic Usage Recorders (TUR), or a maximum of 14,400 inputs, as well as the 1000 inputs already available to the ETDC. Since the ETDC design had a 15-bit word (two start bits, an ICUR bit, ten data bits, a parity bit, and one stop bit) with only ten data bits, a multiword format was necessary. Two words were used to indicate the state of six inputs. The ICUR bit is set in each word, and the remaining 20 data bits are employed as follows. Two bits in one word serve as a first or second word indicator, while the remaining 18 bits indicate the six input states, and the address of those six inputs [two bits for the TUR number (0–3), four bits for the horizontal (0–9), and six bits for the vertical position (0–59)]. These data are intercepted by the scanning process, assembled and buffered, and then used to update individual circuit data, as well as form grouped totals which are merged with the accumulated peg count information. The updating process involves reading the usage and the peg count data from the disk, merging, and then writing the merged peg count data to the disk. The reads and writes are performed using Direct Memory Access (DMA) techniques. Dual-port memory arrangements are needed to accommodate the cumulative load of this processing and the data collection process. The system uses the second port on the ICUR disk as a transfer port, loading data into the added memory port. The individual circuit data are passed downstream and analyzed by the ICAN program described in the article on TNDS equipment systems in this issue.[1]

As memory costs decreased, devices with internal data storage became available. These sources could accumulate data in the central office terminal and forward it to the central unit on command. This method of operation had the potential of a low initial cost for a small number of offices by reducing the initial complexity of the central data collection facility. In addition, there was the opportunity to handle small offices via dial-up data links where dedicated links were not cost effective. For small offices only a few inputs are required, network management data are not needed, and only busy hour information is collected. To serve small offices, the Pollable Data Terminal was designed, and an interface specification published to permit traffic-data-gathering equipment to be connected in this mode. Operation of the Pollable Data Terminal is over a dial-up link, and depending upon busy hours, one EADAS interface port can serve many offices.

4.3 Electronic CU operation

The No. 1A EADAS is designed to collect data from electronic offices only, since plans to replace electromechanical offices were well under way and sufficient No. 1 EADAS Systems would soon be deployed to accommodate these offices until replaced. Initial software was developed by deleting the electromechanical features from the No. 1 EADAS. The initial No. 1A EADAS software was written in assembly language and ran under a very specialized operating system. The No. 1A EADAS software has since been rewritten in C language and now runs under the *UNIX** operating system.

Conversion to the *UNIX* operating system led to several interesting software operations. First is the reception of the accumulated data from the *ESS* switching equipment offices. Each switching machine type has its own unique output format (i.e., 1 *ESS* electronic switching is different from 2 *ESS* switching equipment, from 3 *ESS* switching equipment, etc.). Thus, a unique program was written to receive the data from each type, convert it, and store it in a common form. This task is handled by using LEX, a lexical scanner generation program, and YACC (Yet Another Compiler Compiler) to produce a scanner and parser to process the incoming data stream. The incoming data from the office are treated as a grammar. The description of this grammar is fed to LEX, producing a scanner which identifies the

* Trademark of Bell Laboratories.

tokens. A second description is fed to YACC to generate the parser. The token output from the scanner is passed on to this parser, which then prepares the data for storage by converting the data to binary and removing the header text. This approach significantly reduces the effort necessary to add new office types to the data collection portion of No. 1A EADAS. In addition, if errors are detected in data specification (either the format has changed or the specification was improperly interpreted), modifications are easily made by changing the descriptions of the grammar.

4.4 Overview of the No. 1A EADAS File System

The function of the No. 1A EADAS File System is to provide storage and retrieval capabilities for near-real-time switching network data collected from *ESS* switching equipment offices. These data are needed to produce status and exception reports for network administrators, and for transmission to EADAS/Network Management Systems, as well as downstream data analysis systems. To provide these functions, No. 1A EADAS must store large amounts of data collected from the *ESS* switching equipment in a database. This task is complicated by the high volume of data, the data storage and retrieval speed required of the file system, the large number of offices from which data are collected, and the necessity for storing data on disk over time for use by the report generation programs. The impact these requirements had on the development of the file system for No. 1A EADAS is described in this section.

4.4.1 File system requirements

The No 1A EADAS File System provides the following capabilities:

1. It allows efficient input of near-real-time register data collected from *ESS* switching equipment offices. EADAS is able to collect data simultaneously from at most 48 entities and store it in the database.

2. Many different types of data collected from *ESS* switching equipment offices are stored in the database: discretes, five minute, hourly or half hourly, daily, weekly, etc. The type of data actually stored is a function of the particular office involved, and the number of registers varies by office and type of data. For example, the 1 *ESS* switching equipment may send about 7000 registers every half hour for report generation, but only about 1400 registers for its daily transmissions.

Although register data are by far the largest in volume and require the most rapid input and retrieval, there are several other types of data that are stored in the EADAS database. These include per-office reference data, thresholds for data exception generation, and time schedules for distributing reports to various network terminals connected to the system. The file system accommodates these files as well as the register files. Currently, 36 types of data are stored in the EADAS File System.

3. Most EADAS data are collected on a periodic basis (e.g., every five minutes or every half hour). There are several reasons why successive intervals of data must be stored in the EADAS database. In some cases, calculations are performed on data received during successive collection intervals. In other cases, there is a need to have access to "old" data so that reports may be generated from special studies or for investigating office problems. Finally, a buffering scheme is needed so that reports can be generated simultaneously with data collection.

The number of collection intervals typically varies by type of data. For example, half-hour data are stored for 128 intervals (two and one-half days), whereas daily data are stored for seven days. The file system is able to store different amounts of data for each office and type of data. The total number of files necessary for EADAS to store in its database can be as high as 15,000 to 20,000.

The volume of registers collected varies not only by office and type of data, but also by collection interval. Thus, office A may normally send 2000 registers for data type 1, but at some point may decide to increase the transmission to 2500. Although this event does not occur very often, EADAS is able to accommodate the increase so that no data are lost and does so without increasing file overhead for any other office, data type, or interval. No backup of transient (register) data is necessary. The cost of a rare loss does not justify the cost of regular backups.

4.4.2 The No. 1A EADAS File System

The EADAS File System was implemented using raw (i.e., unbuffered) input/output (I/O) to meet the above requirements. It is a higher-level interface to the Logical File System (LFS), which is used in several other projects, as well as in EADAS. The overall function of the LFS is to subdivide an area of disk into contiguous files and perform all the necessary file management. The following is a list of the most important features of the LFS, as well as some information about how EADAS uses it:

1. Unlike the standard *UNIX** operating system, which provides a hierarchical file system using ASCII path names, LFS provides a "flat" file system. The entire LFS disk area is treated as a single *UNIX* system file that when opened allows access to all logical files. Each file corresponds to a unique number, called a logical file number (lfn),

---

* Trademark of Bell Laboratories.

that is the sole means of referencing the file. Up to 65,535 lfn's per file system are permitted.

2. The LFS performs all space management and low-level I/O, which is transparent to the user. Supported functions are creating, deleting, reading, writing, and switching logical files. LFS disk areas may be mounted and unmounted much like *UNIX* file systems.

3. One of the major functions of the LFS is to perform the mapping from logical file number to physical file location. The LFS requires no opening of individual files since they are "opened" by simply referencing them. Although efficient, access by file numbers is very cumbersome since they reveal little about file contents. Thus, the EADAS File System uses a specialized "open" function to translate retrieval keys (office, type of data, and time) into lfn's. Compared with the *UNIX* file system, the EADAS File System (using the LFS) allows many more open files and avoids costly directory searches in the opening process.

4. All logical files consist of contiguous disk storage. By computing offsets into contiguous files, the LFS avoids the overhead inherent in indirect blocks. Contiguous file storage also allows the LFS to read or write many blocks with one I/O request. This allows the physical block size to be determined by the access requirements for data in a *particular* file rather than be determined arbitrarily.

5. Since the LFS is implemented using the *UNIX* system's raw I/O, data may be transferred directly to/from disk from/to user memory.

In the present No. 1A EADAS configuration, the No. 1A EADAS File System uses three LFSs, with a maximum of 48,000 logical files. Typical EADAS database sizes range from 45M to 70M bytes, depending on the mix of switches from which data are being collected. Thus, the LFS results in efficient storage to permit the No. 1A EADAS to effectively store data from the electronic offices.

Once data are collected, the remainder of the system programs are executed to provide the reports and outputs necessary to administer the network.

V. NORGEN REPORTS

The intent of the Network Operations Report Generator feature is to provide a time-shared programming environment for the development of reporting functions, and a set of reports designed to meet universal reporting needs including near-real-time exception reports. The reports (referred to as Network Operations Reports) are designed to meet the basic reporting needs for each switching entity type served by No. 1 and No. 1A EADAS. These programs provide users with a package that may be modified or supplemented with locally developed features for the support of local reporting needs.

5.1 Basic objectives

Most of the processes that use traffic data operate on data collected over a long period of time, to assure statistical accuracy. Examples of such long-range functions are equipment and facility provisioning, load balancing, and optimized utilization of present equipment and facilities. The reporting functions associated with the above are performed by the "downstream systems" of TNDS. The report results are generally made available to the user within one to several weeks after the end of the study period.

However, there are a number of processes that must be completed in short time ranges. Network performance levels must be monitored for service-affecting problems on a daily basis. Analysis of the traffic data provides valuable insights to the causes for many types of service-affecting problems. This problem-related information is most useful when provided on a near-real-time basis. Furthermore, validation of the data should be done in near-real time to minimize loss of data when something goes wrong. The EADAS Network Operations Reports are designed to meet the near-real-time information needs, while the long-range data collection needs, as indicated earlier, are handled by the TNDS downstream processing systems.

5.2 Report types

The following report types are provided in the generic program:

1. Service Exception Reports are generated when high-priority service faults occur which may require prompt attention or corrective action.

2. Component Exception Reports provide an early warning indication of component conditions which, if allowed to persist, could affect service.

3. Trunk Exception Reports are printed when the percent overflow on final group(s) exceeds an assigned threshold. This report also provides supporting data that indicate where the pressure on the final group(s) is coming from.

4. Summary Reports provide a temporary record of busy-hour central office and trunking conditions. These reports are retained by network administrators until similar data become available from downstream processing.

5. Special Reports provide information which is useful to customers who use centrex, tie line, and multiline hunt-group facilities. The information in these reports is used by Marketing to advise customers

about the performance and needed changes in their communcations services.

6. Daily Reports are generated once a day during the early morning hours. The reports use data that was accumulated by EADAS or added up over the day by the switching system and transmitted daily to EADAS.

7. Network Switching Performance Measurement Plan reports are used to measure central office efficiency. They are organized for easy interpretation, and data are clearly labeled to simplify communications among network administrators, maintenance personnel, and traffic engineers.

8. Division of Revenue Reports provide data that help separate the cost of a switching entity into toll service, local service, and other rate base categories.

9. Measurement Apparatus Exception Reports are used to indicate failure of the data collection devices and associated data links. The report is also used to track corrective action, especially when support from maintenance organizations is required.

5.3 Report users

The principal user of the Network Operations Reports is the Network Administration Center (NAC). The Switching Control Center (SCC) and the Network Data Collection Center (NDCC) also use the reports, but to a lesser degree.

The NAC is responsible for the quality of the local dial service provided by the local central offices and the connecting trunk groups. A major NAC objective is to detect problems as quickly as possible before they can affect service. It is the NAC's responsibility to take the necessary action or make the necessary recommendations to the appropriate organization for correction. These problems may be the result of underprovisioning of equipment or trunks, traffic imbalances, equipment outages, etc.

The NAC is also involved in network transition and cutover activities associated with growth. Real-time analysis of traffic data is a way to ensure that additions and rearrangements work properly.

The SCC has primary responsibility for the equipment maintenance-related aspects of local dial service. Maintenance data such as that contained in the Network Switching Performance Measurement Plan, which is generated by EADAS on a daily basis, are used as an aid in detecting and correcting equipment faults, and in setting maintenance priorities.

The NDCC is responsible for the administration of network and special studies data collection, supervising the operation and maintenance of EADAS, the data links, and data collection apparatus, such as the EADAS Traffic Data Converter, Traffic Usage Recorder, and Pollable Data Terminal. The center is also responsible for coordinating the operation of the TNDS Central Office Equipment (TNDS/EQ) downstream systems. This center is responsible for reviewing measurement Apparatus Exception Reports (AER) and acting to correct problems which cause loss of data.

5.4 Users of special reports

The near-real-time needs of various users continue to evolve. For example, special traffic studies on facilities associated with large business customers have increased substantially in the past few years and all indications are that they will increase even more in the near future. Organizations such as Business Services and Marketing are currently increasing the number of requests for special studies data. These requests are based on increased regulatory and competitive pressures as well as the increased availability of new features made possible by the widespread installation of stored program control systems and a higher level of sophistication of telephone customers. These factors have also combined to increase the frequency of these requests. The study results are often required in almost near-real time, making EADAS the logical system of TNDS to generate them. Studies not required in near-real time, or involving large amounts of data, such as those used for long-range planning, may be more efficiently produced downstream by other TNDS systems, such as TDAS.

The Network Operations Reports deliver the requested information and provide the necessary surveillance for all studies, both near-real time and long range.

5.5 Typical report use

Dial tone speed tests results are part of the Network Operations Reports and are used in the real-time service analysis process. An example of a 1 *ESS* dial tone speed service exception report is shown in Fig. 6. The intent of this report is to alert the network administrator that an abnormal dial tone delay condition has occurred so that corrective action may be taken. This report is divided into four sections: dial tone speed results, customer digit receiver (CDR) data, remote switching system, and supporting data. The first section contains the measured percent dial tone speed results by customer digit receiver type, i.e., *Touch-Tone** dialing and Dial Pulse (DP) with the associated test and delay counts. This report is only generated when the percent *Touch-Tone* dialing or DP dial tone delay exceeds a user-

* Registered service mark of AT&T.

* * * 1A *ESS* SWITCHING EQUIPMENT SERVICE EXCEPTION * * *

ENTITY: XXXXXXXXXXX DATE: 8/11/82 TIME: 12:30 INTERVAL: 30 MIN

* DTS SERVICE EXCEPTION *

	% DTD	TSTS	DLYS
* TT	0.3	363	1
* DP	0.0	87	0

CDR GROUPS	% OFL	OFL	PC	% OCC	# MB	NCI	HT
* COM	0.0	0	0	0.0	0.0	16	
* TT	0.0	0	13916	55.9	8.0	115	7.3
* DP	0.0	0	10646	47.1	11.0	114	11.1

ROB Q PC = ROB OCC = ROB QTY = 0

RSSID = XXXXXXXXXXXX

	% DTD	TSTS	DLYS
* TT/DP	0.0	275	0

ORIG PC = 224 LN-LN USAGE = 467

	% OFL	OFL	PC	% OCC	# MB	NCI	HT
CHANNEL	0.0	0	428	24.0	0.0	108	109.1

E-E CYCLES =	8504	LN SCAN COMP =	8504
ORIG PC =	24401	INC PC =	3594
A LK CCS =	27899		
BLK DT Q PC =	0		
POB %OCC =	20.2	POB OFL =	

*REGISTERED SERVICE MARK OF AT&T

TT – *TOUCH-TONE*® SERVICE

Fig. 6—Example of 1 *ESS* switching system dial tone speed service exception report alerting network administrator of abnormal dial tone delay.

specified threshold. The remaining three portions of the reports provide supporting data for customer digit receivers by type.

When an excessive dial tone delay condition exists, the dial tone delay statistics are printed for both *Touch-Tone* dialing and DP. An asterisk is printed on the line(s) for which the threshold failure(s) have occurred. A threshold test of the CDR group(s) is then performed. If a common CDR group is provided, a threshold test is made for excessive overflow. If the test fails, the results are printed for the common group with an asterisk followed by two lines of information for the DP and *Touch-Tone* CDR groups. Otherwise, the common group results are printed without the asterisk. If a common CDR group is not provided, threshold tests on the DP and *Touch-Tone* dialing CDR groups are made and the results are printed with an asterisk, where required.

After the CDR results are printed, the next section will be provided only if the *ESS* switching equipment is an RSS host office; otherwise this section of the report is omitted.

The last section of the report is always provided and contains overall call processing load data.

Dial tone speed data are analyzed by the Network Administrator following reception of a dial tone speed service exception report. This report is generated when the percent dial tone delay threshold is exceeded for DP and/or *Touch-Tone* dialing. If a common CDR group is provided and it exceeds the percent overflow threshold, then the following analysis would be performed:*

1. The CDR percent occupancy is observed to determine if it exceeds the engineered capacity. If it is too high, then the CDR maintenance busy item (number of circuits made busy) is examined.

2. If the maintenance busy value is too high, then the network administrator requests restoral of the made busy CDRs.

3. If the maintenance busy value is not too high or if there would be a problem even after they are restored, then an analysis of the DP and *Touch-Tone* dialing CDRs is performed. If both the DP and *Touch-Tone* dialing CDR overflows are high and their holding times are unusually long, then permanent signal conditions are analyzed.

4. If holding times are normal and no extraordinary demand exists (such as a severe storm produces), then this condition would be referred to the traffic engineer if it persists.

If *Touch-Tone* dialing percent overflow is high and the DP percent overflow is not, then the DP *Touch-Tone* dialing CDRs should be rebalanced or *Touch-Tone* dialing CDRs added.

If DP and *Touch-Tone* dialing CDR percent overflow is not high, then a check of all related database, circuit quantities, and register assignments should be performed.

The Network Operations dial tone speed service exception report provides the network administrator with the necessary data to support this problem analysis.

In rare cases traffic peaking can be a factor in high CDR overflow. Examination of the 15-minute traffic count report which is generated by the 1 *ESS* switching equipment would provide the necessary 15-minute data required for this analysis to supplement the 30-minute data available to No. 1A EADAS. There are three other areas of investigation related to the dial tone speed analysis. They are Processor Overload, Network Blockage, and Translations. The first two areas would make use of a number of Network Operations Reports in the analysis procedure. The translation analysis would use the dial tone speed service exception report.

VI. NORGEN IMPLEMENTATION

The development of the NORGEN feature was an ambitious under-

---

* Familiarity with Bell System traffic terms is assumed for this section.

taking. Data coming from 48 or more entities are interpreted and analyzed in real time. Formatted reports are prepared and distributed. The internal architectures of a large number of switches determine the report structure and algorithms in fine detail. The program can be distributed to all operating companies and run without local programmer support. Yet the system is tolerant of user programming modifications.

6.1 The challenge

A major challenge was to implement NORGEN in No. 1 EADAS, with a total Random Access Memory (RAM) capacity of 256K bytes (only 64K bytes available to NORGEN), without compromising NORGEN's ultimate capability in the No. 1A EADAS, which has essentially unlimited RAM capacity (the current memory size is 1M byte).

The solution was to use two entirely different software architectures. Yet the feature appears very similar to the users of the two systems.

6.2 NORGEN in No. 1 EADAS

In the No. 1 EADAS, programs are stored in interpretive language to conserve RAM. Program is brought into core in 512-byte blocks and executed block by block. A very linear coding style is used, with a minimum of subroutine calls to hold down the need for calling in extra blocks of code.

The interpretive language (object) is compiled from source code written in K language, which was designed for the project. K language is similar in style to languages with the same level of generality, but has fewer elaborate commands. Thus, K language is easier to learn than C, although C has more "power" in some applications (that is, certain features of C allow some functions to be performed in fewer lines of source code). A text editor and source-to-interpretive language compiler, both written in assembly language, are supplied with the generic.

In K language, all variables reside logically on disk; there is no distinction between local and global variables. The system keeps a cache of disk blocks in memory to improve the speed of operation. This structure was chosen because of the small amount of memory available, but has been found to ease coding problems because the programmer need not consider whether a variable is of one type or another.

The disk file access has three levels. A directory contains all named files and file offsets for particular data items. The directory is accessed using a backup function. When an item is needed, a fast access directory is checked first, to see if the file is already in RAM; then the backup function is used to access the disk directory; and then the file itself is called from disk. Most requests are found in RAM; nearly all others are filled by two disk accesses (one for the directory and one for the actual data requested).

6.3 NORGEN in No. 1A EADAS

In the No. 1A EADAS, programs are stored in assembly language, compiled from C language source code by the *UNIX* compiler. This permits more subroutine calls and a more conventional programming style. The *UNIX* text editor and C language compiler are provided with the generic.

The *UNIX* file system is used for program files and parameters, but the specialized two-level Logical File System as previously described is provided for accessing the traffic data. Otherwise, multiple file accesses would be required for the multilevel directory structure of *UNIX*.

6.4 Parameter administration

In this article, the term "parameters" refers to all the user-controlled tables which the generic program reads to relate to local conditions. NORGEN parameters include scheduled times for reports to be run, exception thresholds, register definitions, etc. Any parameter administration system must have the functions add, delete, change, display, and backup.

In both the No. 1 EADAS and No. 1A EADAS, these functions are served by using text files, in controlled format, to hold all the data. The text editors, which are quite similar, are used for all of the above functions except backup; this is accomplished by dumping all the parameter files onto tape (approximately weekly) and reading them from tape in an emergency.

Programs are provided to convert each of the text files, by type, into binary object files for efficient access. When any text file has been changed (by the editor), the file is recompiled.

The text editor enables the user to work in a relatively English-like environment, minimizing training and making it easy to spot errors by browsing displays of the files.

6.5 Raw register access

The No. 1 and No. 1A EADAS use very different register access mechanisms. Both use key words to provide mnemonic access to individual or summed groups of registers. However, the No. 1 EADAS systematically translates the packed register data from data collection into rigidly structured tables, with each key word assigned a fixed location (for each switch type). On the other hand, the No. 1A EADAS uses mapping algorithms and parameter tables to extract each key word from the raw registers as it is called for from the program. This newer technique provides a great savings in disk space, allowing a fourfold increase in the number of intervals of data that can be stored, since most entities have only a minority of the total possible key words actually in use. The real time used by the two methods appears to be similar. The older technique is more efficient when key words are used repeatedly, but very few key words are used more than once or twice.

6.6 *Scheduler*

Again, there are strong differences between the two schedulers. In the No. 1 EADAS, the scheduled reports are run one at a time, to minimize the swapping of files into and out of the severely limited RAM space. In the No. 1A EADAS, two modules are run at one time (level 2 multiprocessing). This achieves an efficiency gain due to interleaving processor and disk I/O functions.

Both executives are time-shared, allowing user access to demand reports, dump key-word values, etc. The No. 1 EADAS executive allows new users high priority. As recent run time increases for a given user, that user's priority (probability of being run at the next opportunity) decreases. This automatically allows users high priority for short, easy tasks like editing or key-word dumps while lowering the priority for compilations and running long reports.

The No. 1A EADAS uses the *UNIX* priority feature, giving manual users priority over scheduled work, independent of their tasks.

6.7 *Reports*

The report modules are written in C language, using straightforward, linear style. This facilitates user programming as well as Bell Laboratories and Western Electric modifications of the design. Subroutines are used sparingly, so that calculation details are grouped in one place in the source program.

Special functions are defined for pervasive features. In particular, a threshold check function automatically seeks the threshold parameter and applies the appropriate check (less than, equal to, range). A special print function prints a blank in a field when appropriate. (For example, data pertaining to items which are unequipped in a given entity are carried in memory as negative numbers, which are suppressed by the print function.)

Standard page headers and section header routines are called to provide uniform structure of the reports. Automatic pagination is provided (through the special print function).

The layout of the reports emphasizes the grouping of related data for easy reading. White space is used freely to set off the groups.

Mnemonic labels are provided; abbreviations are chosen to be consistent with related Bell System documentation.

6.8 Report distribution

After the ASCII image of a report is delivered to the spooling system, it is distributed to appropriate user terminals. Each report has a message class number. A parameter table lists the destination terminals for each entity and each class. In No. 1 EADAS, this table was centrally administered; in No. 1A EADAS, the user at each terminal designates the entities and message classes it is to receive. A particular report thus may be sent to many terminals.

VII. TAPE WRITING

In the No. 1 EADAS, tape writing is done by an assembly-language-coded program that runs under the data collection executive. Data are written each half hour, under control of schedule parameters. The data are recorded in ASCII characters; each register is represented by a six-digit number.

In the No. 1A EADAS, tape data are prepared in two steps. The first step is to process and format the data; the Tape Data Formatter (TDF) module runs under the NORGEN executive. This provides opportunity for much more powerful processing of the data. In particular, the TDF combines two successive half-hour data messages into a single one-hour message. In most cases, registers from each half hour are added together; but certain registers, such as those containing averages, peaks, or trunk group numbers, are combined by specialized algorithms. Improved validity checking is done to avoid passing bad data downstream.

In the second step, the TDF writes the formatted data in a spooling area, where they are held for up to several hours. This spooling capability improves operational flexibility; for example, it is possible to hold data while a tape drive is being repaired.

A new tape format is provided in No. 1A EADAS. Data are usually written as 16-bit binary words, thus fitting a register into two bytes, where the No. 1 EADAS requires six. The effective reduction is approximately 2 to 1 since there is less compression in headers, and no compression of record gaps.

Industry standard tape labels in the new format facilitate data-link transmission of tapes and inventory control.

Two tape drives may be provided in the No. 1A EADAS; when one tape is filled, the other begins to write. This feature, together with the binary format, combination of half-hour data into one-hour data, and spooling, allows data collection to continue unattended in most installations over long holiday weekends.

VIII. DATA FLOW

Following the course of data through the No. 1A EADAS shows how major program modules relate to one another. Let us follow the data of and *H* schedule* from a polled 1A *ESS* switching equipment office—the schedule for the interval from 10:30 a.m. to 11:00 a.m., in particular.

The 1A *ESS* switching equipment office will have collected the data and made it ready for transmission shortly after 11:00 a.m. on its clock. The No. 1A EADAS will wait several seconds after 11:00 a.m. on its clock to allow for system time differences.† Then it will send a poll asking for the first block of 256 registers of *H* schedule data. When the block has successfully been received, it will ask for the next block, and so on, until the proper number of registers (a channel parameter) has been validly received.

As the blocks of data are received, they are stored in the logical file system. A new entry has been made in the LFS directory, so that the data may be accessed by entity name, end time of data interval, and type of data.

When the schedule is complete in the LFS, a message is sent via a pipe to the NORGEN Data Analyzer (NDA) executive. The executive checks schedule parameters to see which program modules are to run on this particular *H* schedule.

Suppose that the busy hour for this entity is 10:00 a.m. to 11:00 a.m. Then the Load-Service Summary module will be scheduled among others. The executive will make an entry in its job queue, listing entity, module, and end time of data. When this entry works its way to the head of the first-in-first-out queue and a new process can be started, the executive will run it. This particular module (the Load-Service Summary) works on a full hour's worth of data. Therefore, it retrieves two successive intervals of data from the LFS and adds their register values together to form an equivalent one-hour schedule.

Next, the module accesses data to prepare its report. For example, it reads the peg count of incoming calls by use of key word INCPC (incoming calls peg count) as a variable in a program. The compiler has recognized this key word as a register reference and has compiled a function call to the key-word access routine. The key-word access routine references the Data Collection Device (DCD) parameter table to find the register number and return the corresponding value to the Load-Service Summary module.

The module prepares the ASCII text of the report, calling special

* An "*H* schedule" is a set of registers that represent traffic through the common parts of the switch (as opposed to the trunk groups).

† The No. 1A EADAS periodically requests time-of-day from each 1A *ESS* switching equipment office; if the time difference is large, a message is printed.

subroutines to form the header, subsection header, and end of report headers. Pagination is provided automatically; to make this possible, a PRINTN function is defined, which performs special NORGEN report functions before calling the PRINT functions.

The ASCII text is stored in a spooling area, but the spooler directory entry is not completed until all modules for that entity and interval are completed. This allows the NDA executive to mark the sequence of reports in a fixed order, so that they will always appear in the sequence expected by the user. This facilitates search and filing of specific reports.

The report spooling program distributes each report in order. Each report type is assigned a message class. The terminal tables are searched to determine which terminals have designated the entity name and message class for the Load-Service Summary Report. The report is sent to these terminals through the multiplexor driver program.

IX. CAPACITY

In the No. 1 EADAS, NORGEN is usually the limiting capacity consideration. Consequently, the capacity characteristics of the reports have been analyzed carefully.

The capacity criterion of the system was to complete the report generation load of an "equivalent half hour" in one half hour, to prevent excessive delays in the availability of reports. The concept of an "equivalent half hour" was introduced because the report load is irregular. For example, for a polled interface, some reports run continuously (every half hour), some run every hour, and some only after busy hours. After discussions with users, it was decided that a backlog can build up during a busy hour, as long as the system recovers within one and one-half hours.

The load on the system for an equivalent half hour was taken to be:

continuous reports: fully weighted
hourly reports: one-half weighted
busy hour reports: one-third weighted.

The capacity analysis was thus reduced to estimating the run time of each report module. The weighted run times are then added to determine whether the system can keep up with its load.

Report run times are composed of two major components; processor time and disk access time. Each of these resources is subject to priority seizure by data collection activities, which extends the report run times significantly. Extensive modeling and measurement of the No. 1 EADAS and No. 1A EADAS Systems have been performed to determine the quantitative effects of data collection and relate those to the mix of sizes and types of remote data sources (switching entities).

9.1 Modeling

The application of analytic modeling to combination real time and time shared systems has been challenging. It has been successful because a limited accuracy goal of plus or minus five percent was set. As a result, effects which sum to only a few percent were neglected, or assigned a fixed allocation which was neither best nor worst case, but somewhere in between.

Attention was thus directed at the first-order effects. Measurements were carried out in the laboratory, using a field configuration system under approximately 50-percent load. The load was applied by a separate minicomputer acting as a load test generator. Traffic data used was based on data collected from "typical" offices in the field. These model offices were chosen to be reasonably complex in terms of the number and types of registers, but worst-case offices were avoided since they do not dominate a fully loaded system.

By taking measurements on a system loaded in the 50-percent range, the effects of approximations in the model are minimized.

9.1.1 No. 1 EADAS capacity model

In the No. 1 EADAS, the scheduled report modules are run sequentially, rather than multiprocessed, because of the limited availability of RAM. In this system, data collection takes a large amount of processor time at interrupt level (up to 60 percent in a heavily loaded system), especially for the ICUR feature. The effect of this data collection load on a given report module depends on how much processor time it needs; the data collection load has no first-order effect on disk access time. The effect, then, is to extend the processor time component of the report run time:

$$T_p = \frac{T_{po}}{1 - F},$$

where:

T_p = processor component of the report run time in the presence of data collection load,

T_{po} = processor component of the report run time in the absence of data collection load,

F = fraction of processor used by data collection.

In the above model, it is important to note that the data collection work is not time shared with the report modules, but is performed at priority interrupt.

A second-order effect, which is typically large enough to be significant (of the order of ten percent), is caused by base-level data collection load. This load has no effect on processor time because it always takes place during times when report modules are waiting for disk accesses. However, when a disk access is complete, base-level work is usually in progress and neither a further disk access nor a report processor segment can be started until it completes. Effectively then, a fraction (taken to be 1/2) of the base-level data collection time (stretched out by interrupt data collection time) is added to disk latency.

$$L = L_o + \frac{1}{2}\frac{a}{1 - F},$$

where:
L = disk latency in the presence of data collection load,
L_o = disk latency in the absence of data collection load,
a = average base-level time for data collection,
F = fraction of processor used by data collection at interrupt.

The effect on the disk portion of the report run time is given by:

$$T_d = T_{do}\frac{L}{L_o},$$

where:
T_d = disk component of the report run time in the presence of data collection load,
T_{do} = the disk time measured in the absence of data collection load.

The rule for capacity estimation is to add the processor and disk component times for the relevant report modules (for the given mix of office type and report schedules, according to local operation practices) weighted by the appropriate factor according to the report frequency, to apply the model equations to account for the estimated data collection loads, and to compare the resultant time to the 1800 seconds of each half hour. Ten percent of the report capacity is allocated to allow for manually generated load running in time share with the reports.

While the above process may seem complex, it has been reduced to a straightforward application of tables and worksheets and has been well accepted by the operating companies who must estimate their capacity.

9.1.2 No. 1A EADAS capacity

The No. 1A EADAS is quite similar in its modeling except that only one effect of data collection load is significant—priority access to the disk. Since scheduled reports are run in time share, two at a time, and since the reports need about 60-percent of disk I/O time and only 40-percent of processor time, the reports are strongly disk limited. Measurements show that about 85 percent of the disk is utilized, which is quite constant even when 50 percent of the disk is taken by data collection. A change in data collection architecture has resulted in heavy disk loads for collection of ASCII data in the No. 1A EADAS.

The system is found to be so disk limited, that the distinction between processor and disk components can be neglected, and the *effective* run time (which is measured total run time in multiprocessing, divided by two, since two reports are run at the same time) can be considered to be all disk time.

Then the effect of data collection disk load is given by:

$$T = \frac{T_o}{1 - F},$$

where:

T = effective report run time in the presence of data collection load,
T_o = effective report run time in the absence of data collection load,
F = fraction of the disk used by data collection load.

Estimation of capacity is then carried out in essentially the same manner as for the No. 1 EADAS.

X. SUMMARY

EADAS has met its original operational objectives by providing for the collection of network data for engineering, network managment, and real-time data analysis. The result has been an overall improvement in the monitoring of network operations, thus meeting the increasing need for more efficient network utilization to optimize both capital investment and customer service. Eliminating the manual effort formerly associated with the collection and processing of network data has resulted in substantial economic savings. EADAS will continue to be enhanced to provide interfaces with new switching machines and operations systems.

XI. ACKNOWLEDGMENTS

The No. 1 and No. 1A EADAS represent the work of many people over a full decade. The features of the system are reported here without specific accreditation, which would be difficult at this time. However, the authors wish to acknowledge the special help of J. R. McSkimin and R. L. Hardin in the preparation of this paper.

REFERENCES

1. N. D. Fulton, J. J. Galiardi, E. J. Pasternak, S. A. Schulman, and H. E. Voigt, "Total Network Data System: Equipment Systems" B.S.T.J., this issue.

AUTHORS

J. A. Grandle, Jr., B.E.E., 1967, University of Virginia; M.S.E.E., 1968, Purdue University; Bell Laboratories, 1967—. Mr. Grandle has helped design Traffic Measurement Circuits; hardware and software for No. 1 Engineering and Administrative Data Acquisition System (EADAS), No. 1A EADAS, EADAS Individual Circuit Recording, and EADAS Network Operations Report Generators; CDA; No. 1 Automatic Message Accounting Recording Center central hardware; PDT-2A hardware and firmware; and processor terminals, including the PLURAL multiprocessor terminal. He also has three patents and has published several articles. ("A New Test Line for Coin Phones," and "A Versatile Data Gathering Tool") published in the *Bell Laboratories Record.*

Donald J. Gagne, B.E.E., 1943, Pratt Institute; Bell Laboratories, 1943—. Mr. Gagne has participated in the development of order wire and alarm facilities, mobile and overseas control terminals, and No. 1 and No. 5 Crossbar circuits. He has also conducted systems engineering studies for new recorded announcement systems, new electromechanical and electronic PBXs, *PICTUREPHONE*® meeting service and Automatic Number Identification for PBXs, and new Service Observing and Traffic Measurement facilities. He supervised early Systems Engineering for EADAS and at present is supervising a group responsible for EADAS Systems Engineering.

George H. Wedemeyer, Certificate in Aircraft Design and Technology, 1959, Academy of Aeronautics; Bell Laboratories, 1962–1981; AT&T, 1981—. At Bell Laboratories Mr. Wedemeyer participated in the hardware and software design of data collection devices. In 1981, he became a Member of Technical Staff while working for the Network Administration Systems Department in Holmdel, New Jersey. His responsibilities included formulating requirements for traffic-data collection and reporting systems, economic analysis, and system evaluations. Currently, Mr. Wedemeyer is a staff manager at AT&T responsible for the project management of EADAS.

Charles J. Byrne, B.S.E.E., 1957, Renesselear Polytechnic Institute; M.S.E.E., 1958, California Institute of Technology; Bell Laboratories, 1958–1962, 1968.— Before rejoining Bell Laboratories, Mr. Byrne was employed at Bellcomm, Incorporated, where he worked on systems engineering for the Apollo Project. At Bell Laboratories he has worked in communications systems research and operations systems development. He is currently Supervisor of the International Sales Planning Group of the Switching Operations Systems Laboratory. Member, Sigma Xi, Tau Beta Pi, Eta Kappa Nu.