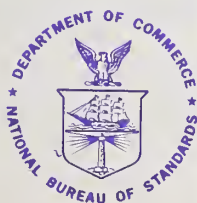


**COMPUTER SCIENCE & TECHNOLOGY:**



# **SURVEY OF REMOTE TERMINAL EMULATORS**



**NBS Special Publication 500-4**  
**U.S. DEPARTMENT OF COMMERCE**  
**National Bureau of Standards**

## NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards<sup>1</sup> was established by an act of Congress March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau consists of the Institute for Basic Standards, the Institute for Materials Research, the Institute for Applied Technology, the Institute for Computer Sciences and Technology, the Office for Information Programs, and the Office of Experimental Technology Incentives Program.

**THE INSTITUTE FOR BASIC STANDARDS** provides the central basis within the United States of a complete and consistent system of physical measurement; coordinates that system with measurement systems of other nations; and furnishes essential services leading to accurate and uniform physical measurements throughout the Nation's scientific community, industry, and commerce. The Institute consists of the Office of Measurement Services, and the following center and divisions:

Applied Mathematics — Electricity — Mechanics — Heat — Optical Physics — Center for Radiation Research — Laboratory Astrophysics<sup>2</sup> — Cryogenics<sup>2</sup> — Electromagnetics<sup>2</sup> — Time and Frequency<sup>2</sup>.

**THE INSTITUTE FOR MATERIALS RESEARCH** conducts materials research leading to improved methods of measurement, standards, and data on the properties of well-characterized materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; and develops, produces, and distributes standard reference materials. The Institute consists of the Office of Standard Reference Materials, the Office of Air and Water Measurement, and the following divisions:

Analytical Chemistry — Polymers — Metallurgy — Inorganic Materials — Reactor Radiation — Physical Chemistry.

**THE INSTITUTE FOR APPLIED TECHNOLOGY** provides technical services developing and promoting the use of available technology; cooperates with public and private organizations in developing technological standards, codes, and test methods; and provides technical advice services, and information to Government agencies and the public. The Institute consists of the following divisions and centers:

Standards Application and Analysis — Electronic Technology — Center for Consumer Product Technology: Product Systems Analysis; Product Engineering — Center for Building Technology: Structures, Materials, and Safety; Building Environment; Technical Evaluation and Application — Center for Fire Research: Fire Science; Fire Safety Engineering.

**THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY** conducts research and provides technical services designed to aid Government agencies in improving cost effectiveness in the conduct of their programs through the selection, acquisition, and effective utilization of automatic data processing equipment; and serves as the principal focus within the executive branch for the development of Federal standards for automatic data processing equipment, techniques, and computer languages. The Institute consist of the following divisions:

Computer Services — Systems and Software — Computer Systems Engineering — Information Technology.

**THE OFFICE OF EXPERIMENTAL TECHNOLOGY INCENTIVES PROGRAM** seeks to affect public policy and process to facilitate technological change in the private sector by examining and experimenting with Government policies and practices in order to identify and remove Government-related barriers and to correct inherent market imperfections that impede the innovation process.

**THE OFFICE FOR INFORMATION PROGRAMS** promotes optimum dissemination and accessibility of scientific information generated within NBS; promotes the development of the National Standard Reference Data System and a system of information analysis centers dealing with the broader aspects of the National Measurement System; provides appropriate services to ensure that the NBS staff has optimum accessibility to the scientific information of the world. The Office consists of the following organizational units:

Office of Standard Reference Data — Office of Information Activities — Office of Technical Publications — Library — Office of International Standards — Office of International Relations.

<sup>1</sup> Headquarters and Laboratories at Gaithersburg, Maryland, unless otherwise noted; mailing address Washington, D.C. 20234.

<sup>2</sup> Located at Boulder, Colorado 80302.

1 1977  
ACC,  
57  
500-4  
19  
2

# COMPUTER SCIENCE & TECHNOLOGY: Survey of Remote Terminal Emulators

E special publication no. 503-4

Shirley Ward Watkins and Marshall D. Abrams

Institute for Computer Sciences and Technology  
National Bureau of Standards  
Washington, D.C. 20234



U.S. DEPARTMENT OF COMMERCE, Juanita M. Kreps, Secretary

Dr. Betsy Ancker-Johnson, Assistant Secretary for Science and Technology

45  
60  
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Acting Director

Issued April 1977

## Reports on Computer Science and Technology

The National Bureau of Standards has a special responsibility within the Federal Government for computer science and technology activities. The programs of the NBS Institute for Computer Sciences and Technology are designed to provide ADP standards, guidelines, and technical advisory services to improve the effectiveness of computer utilization in the Federal sector, and to perform appropriate research and development efforts as foundation for such activities and programs. This publication series will report these NBS efforts to the Federal computer community as well as to interested specialists in the academic and private sectors. Those wishing to receive notices of publications in this series should complete and return the form at the end of this publication.

### National Bureau of Standards Special Publication 500-4

Nat. Bur. Stand. (U.S.), Spec. Publ. 500-4, 80 pages (Apr. 1977)

CODEN: XNBSAV

#### Library of Congress Cataloging in Publication Data

Watkins, Shirley Ward.

Survey of remote terminal emulators.

(NBS special publication ; 500-4 : Computer science & technology)

Bibliography: p.

Supt. of Docs. no.: C 13.10:500-4

1. Emulators (Computer programs) 2. Computer terminals—Remote terminals. 3. Electronic digital computers—Evaluation. I. Abrams, Marshall D., joint author. II. Title. III. Series: United States. National Bureau of Standards. Special publication ; 500-4 : Computer science & technology.

QC100.U57 no. 500-4 [QA76.6] 602'.1s [001.6'4] 77-608013

U.S. GOVERNMENT PRINTING OFFICE  
WASHINGTON: 1977

---

For sale by the Superintendent of Documents, U.S. Government Printing Office  
Washington, D.C. - 20402 - Price \$1.45

Stock No. 003-003-01756-6

## PREFACE

This report identifies Remote Terminal Emulators by trade name for the purpose of identification; inclusion of a system in this report in no way implies a recommendation or endorsement by the National Bureau of Standards. Similarly, the omission of a system does not imply that its capabilities are less than those of the included systems. The information presented was obtained primarily from vendors' documents and/or conversations with vendors' representatives and was reviewed by each vendor for clarity and accuracy as of September 1976; the authors retain final technical judgment on the information included. The presentation should not be construed as a certification that any system provides the indicated capabilities. This report is intended to be informative and instructive concerning remote terminal emulation; it is not an evaluation of the emulators described.





# CONTENTS

	Page
1. Introduction	1
1.1 Historical perspective on computer usage	2
1.2 Historical perspective on measurement and evaluation techniques	3
2. Terminology	5
2.1 Remote terminal emulation	5
2.2 Determinism of SUT's	6
2.3 Scenario	7
2.4 Script	8
2.5 Scene	10
2.6 Integrity confirmation	10
3. Applicability	12
4. Implementations	13
4.1 Air Force/MITRE's Remote Terminal Emulator	17
4.1.1 General description	17
4.1.2 Hardware considerations	18
4.1.3 Script preparation	19
4.1.4 Data acquisition	20
4.1.5 Data summarization	20
4.1.6 Integrity confirmation	21
4.1.7 Current usage	22
4.1.8 Operational considerations	22
4.2 Burroughs' Data Communications Traffic Emulator (System/DCEM)	23
4.2.1 General description	23
4.2.2 Hardware considerations	24
4.2.3 Script preparation	25
4.2.4 Data acquisition	26
4.2.5 Data summarization	26
4.2.6 Integrity confirmation	26
4.2.7 Current usage	27
4.2.8 Operational considerations	27
4.3 Control Data Corporation's Project BARTER	27
4.3.1 General description	27
4.3.2 Hardware considerations	27
4.3.3 Script preparation	28
4.3.4 Data acquisition	29
4.3.5 Data summarization	29
4.3.6 Integrity confirmation	29
4.3.7 Current usage	29
4.3.8 Operational considerations	30
4.4 Digital Equipment Corporation's Script Machine	30
4.4.1 General description	30
4.4.2 Hardware considerations	31
4.4.3 Script preparation	32
4.4.4 Data acquisition	32
4.4.5 Data summarization	32
4.4.6 Integrity confirmation	33

	Page
4.4.7 Current usage	33
4.4.7.1 The IAS experiment	33
4.4.7.2 The data base management experiment	33
4.4.8 Operational considerations	34
4.5 Hewlett Packard's Timesharing Event Performance Evaluator (TEPE)	35
4.5.1 General description	35
4.5.2 Hardware considerations	35
4.5.3 Script preparation	36
4.5.4 Data acquisition	36
4.5.5 Data summarization	36
4.5.6 Integrity confirmation	36
4.5.7 Current usage	37
4.5.8 Operational considerations	37
4.6 Honeywell's DATUS	37
4.6.1 General description	37
4.6.2 Hardware considerations	38
4.6.3 Script preparation	39
4.6.4 Data acquisition	39
4.6.5 Data summarization	39
4.6.6 Integrity confirmation	39
4.6.7 Current usage	39
4.6.8 Operational considerations	39
4.7 Honeywell's Communications User Emulated System for Traffic Analysis (CUESTA)	40
4.7.1 General description	40
4.7.2 Hardware considerations	41
4.7.3 Script preparation	41
4.7.4 Data acquisition	41
4.7.5 Data summarization	42
4.7.6 Integrity confirmation	42
4.7.7 Current usage	42
4.7.8 Operational considerations	42
4.8 IBM's Data Base/Data Communication Driver System (DB/DC Driver)	43
4.8.1 General description	43
4.8.2 Hardware considerations	44
4.8.3 Script preparation	45
4.8.4 Data acquisition	45
4.8.5 Data summarization	46
4.8.6 Integrity confirmation	46
4.8.7 Current usage	46
4.8.8 Operational considerations	46
4.9 IBM's Teleprocessing Network Simulator (TPNS)	46
4.9.1 General description	47
4.9.2 Hardware considerations	47
4.9.3 Script preparation	48
4.9.4 Data acquisition	48
4.9.5 Data summarization	48
4.9.6 Integrity confirmation	49
4.9.7 Current usage	49
4.9.8 Operational considerations	49



	<u>Page</u>
4.10 Tymshare's MEDLINE Simulated User System (MSUS)	49
4.10.1 General description	50
4.10.2 Hardware considerations	50
4.10.3 Script preparation	51
4.10.4 Data acquisition	51
4.10.5 Data summarization	51
4.10.6 Integrity confirmation	52
4.10.7 Current usage	52
4.10.8 Operational considerations	53
4.11 Univac's Communications Network Emulator (CNE)	53
4.11.1 General description	53
4.11.2 Hardware considerations	53
4.11.3 Script preparation	54
4.11.4 Data acquisition	54
4.11.5 Data summarization	55
4.11.6 Integrity confirmation	55
4.11.7 Current usage	55
4.11.8 Operational considerations	56
4.12 Univac's Communications Simulator (CS1100)	56
4.12.1 General description	56
4.12.2 Hardware considerations	57
4.12.3 Script preparation	58
4.12.4 Data acquisition	58
4.12.5 Data summarization	58
4.12.6 Integrity confirmation	59
4.12.7 Current usage	59
4.12.8 Operational considerations	59
5. Problem areas	60
6. Summary	62
References	63

## FIGURES

Figure	<u>Page</u>
1. Remote Terminal Emulation	6
2. Sample Scenario	8
3. Scenario to Script Process	9
4. Scenario to Script Translation	9

## TABLES

Table	<u>Page</u>
1. RTE Implementation Hardware	15
2. Systems Tested	16

## APPENDICES

Appendix	<u>Page</u>
A. Communications Disciplines Supported	66
B. Maximum Number of Devices Emulated	67
C. Emulated User Features	68
D. Measures of SUT Performance	69
E. Data Summarization	70
F. RTE Verification Procedures	71

# Survey of Remote Terminal Emulators

Shirley Ward Watkins  
Marshall D. Abrams

## ABSTRACT

This report describes twelve Remote Terminal Emulators (RTE's). The key terminology associated with remote terminal emulation is defined and possible application areas are discussed. Technical implementation details and operational considerations are addressed for each RTE. Summary tables are provided to indicate current RTE capabilities and capacities, as claimed by the RTE developers.

Key words: evaluation; interactive; measurement; performance evaluation; performance measurement; remote terminal emulation; Remote Terminal Emulators; teleprocessing.

## 1. Introduction

Remote terminal emulation is an approach to the test and evaluation of teleprocessing computer systems in which a driver is implemented external to and independent of the system being tested. The driver, called a Remote Terminal Emulator (RTE), exerts a specified teleprocessing workload on the system such that it appears to the System Under Test (SUT) that it is connected to live terminal devices. Examples of devices emulated include asynchronous interactive keyboard terminals, synchronous interactive terminals, and synchronous batch stations.

This technical report surveys currently implemented Remote Terminal Emulators. Brief historical perspectives on computer usage and corresponding approaches to determining performance are provided. The key terminology associated with remote terminal emulation is defined and possible application areas discussed. Twelve Remote Terminal Emulators are described followed by a discussion of application-oriented problem areas. The appendices summarize certain features or capabilities for all the implementations.

## 1.1 Historical perspective on computer usage

Computers which entered the marketplace in the early 1950's restricted processing to only one program or job at a time. The person/computer relationship was one-to-one as the user interacted with the computer from the system console. This relationship was geared toward a machine with minimal consideration for the human attempting to use it as a tool. The development of an operating system was a major improvement in the person/computer relationship because some of the general management tasks, such as data conversion routines and input/output procedures were assumed by the computer. At this time the structure of the person/computer relationship also changed. Typically programs, data, and run cards were prepared by programmers, also called users of the computer. Information was key-punched on cards in a given format and submitted to an operator at the computer. The operator was responsible for loading the cards into the card reader and collating the output from the user's job. Now instead of one person submitting a job to a computer, many were doing so simultaneously. The technique of grouping jobs prior to processing is referred to as batch processing.

In batch processing the time between submitting a job, having it processed, and obtaining the results is dependent upon many variables; but in such an operation it is expected that the user may wait several hours, perhaps overnight, before knowing the results of the submitted job. Upon returning for whatever output is expected, it is not unreasonable to assume that the user is greeted with the unexpected -- a program error. The error must then be found and corrected, cards repunched and organized, and the entire process begun again. Such usage of computers has predominated over the last two decades.

Moving batch input/output devices to locations more convenient to the computer users was the next development in the usage of computer systems. These devices could reside in a room next to the computer or miles from the computer. Due to the physical location of the batch devices, this type of processing is called remote batch.

Over the last decade another type of person/computer relationship evolved reminiscent of the usage of the early 1950's. Once again the user appears to be in a one-on-one situation with the computer; however, instead of sitting at the console or at a terminal adjacent to the computer, the user may be sitting at a terminal tens or hundreds of miles away. The person/computer relationship is dependent upon a communications facility, and while it may appear to the users that the computer is theirs alone, in actuality many other users are sitting at terminals and communicating with



the computer under the same illusion. This technique is called time sharing because intervals of computer time are apportioned among the users. Now instead of submitting decks of cards to an operator, the user enters work directly to the computer via the keyboard of a terminal. Instead of waiting hours for the results of a computation, the user obtains results in seconds or minutes. Due to the appearance of the person/computer relationship, this mode of usage has been termed interactive or conversational.

The term teleprocessing denotes information handling in which a data processing system utilizes communications facilities. It includes the previously defined interactive and remote batch usage of computer systems.

## 1.2 Historical perspective on measurement and evaluation techniques

The performance of the early computers was typically evaluated by one of two methods. The scientific-oriented computer systems were evaluated in terms of individual arithmetic instruction speed. Dependent upon the calculations to be performed by the computer, the various arithmetic instructions were weighted and approximation of the time required for the average calculation was made based on the instruction speed. The "fastest" instruction set for that application was the determining factor in system evaluation. The performance of commercial "data processing" computer systems was viewed from the opposite end of the spectrum. Processing time was virtually ignored. Rather, computer systems were assessed on their input/output characteristics: e.g. the record reading and writing rate, the cards per minute read/punched, and the lines per minute printed. (Drummond[1973])

A logical measure of the performance of a batch processing computer system is the number of jobs executed per unit time. This measure is commonly called throughput. Hardware components, such as the Central Processing Unit (CPU) and peripheral devices, and software components, such as the scheduling algorithm and interrupt handling, are monitored to determine utilization and performance. The elapsed time it takes to process one particular job is not a major concern because the users are not normally waiting in the computer room. An added delay of a few minutes or hours does not reflect upon the overall system performance if delaying that job enables the execution of many other jobs.

In interactive systems, performance evaluation is augmented by a concern for the processing of individual jobs. The user is of foremost concern in determining the



level of performance of the computer system. Abrams and Cotton[1975] suggest that such measures as response time, availability, accessibility and reliability are pertinent to determining the performance of a teleprocessing system. Response time is frequently defined as the elapsed time from the end of a user's input to the beginning of the system's corresponding message. Availability refers to the interval of time that the computer system is up and capable of performing functions. Accessibility is the counterpart of availability; it included an individual user's ability to access the resources of the computer system. Reliability refers to the computer system staying up and the ability of the computer to function with a minimum number of errors.

These user-oriented measures have required the development of new techniques for determining the performance of computer systems. Remote terminal emulation, as described in the next section, is one of these new techniques to determine the performance of interactive, teleprocessing computer systems.

## 2. Terminology

This section defines key terminology related to remote terminal emulation which is used throughout this report. For the most part these definitions are the product of discussions with many groups, including implementors of remote terminal emulators. The purpose is to establish a common ground for the technical descriptions presented herein.

### 2.1 Remote terminal emulation

Remote terminal emulation is an approach to the performance evaluation of teleprocessing systems in which a driver external to and independent of the System Under Test (SUT) connects to the SUT through its communications device interfaces, either locally or through a communications network, and interacts with the SUT as if the driver were a set of terminal devices and operators. The communication protocols of the normal teleprocessing system are used. In fact, the SUT should be unable to distinguish between communicating with the driver, and with real users and terminal devices. Integral to this technique is a monitor which captures data descriptive of the driver/SUT interaction. Through analysis of this data, performance determinations are made.

A Remote Terminal Emulator (RTE) is a specific implementation of a teleprocessing workload driver employed in remote terminal emulation. A variety of RTE's are described in this report and, as will be observed, RTE's are implemented on various sizes of machines -- from minicomputers to large scale computers. As specified in this report, an RTE must be able to replace the entire terminal workload on a SUT. The RTE must therefore be capable of emulating all, or any part, of the teleprocessing workload. Due to this definition, there is a class of external drivers which are not included in the classification of RTE. Such devices as the Rand Monitor/Stimulus-Generator (see Lockett and Bell[1975] and Yoshimura[1975]) do not meet this definition in that they are limited in the workload which can be emulated.

A monitor to record selected events is a required component of the remote terminal emulation process. Most RTE's incorporate this monitor component; however, a co-resident monitor is not a requirement for a device to be classified as an RTE. It is also possible that an RTE may reduce the data collected by the monitor, but this function may also be performed by another computer. Figure 1 demonstrates the components of the remote terminal emulation process.

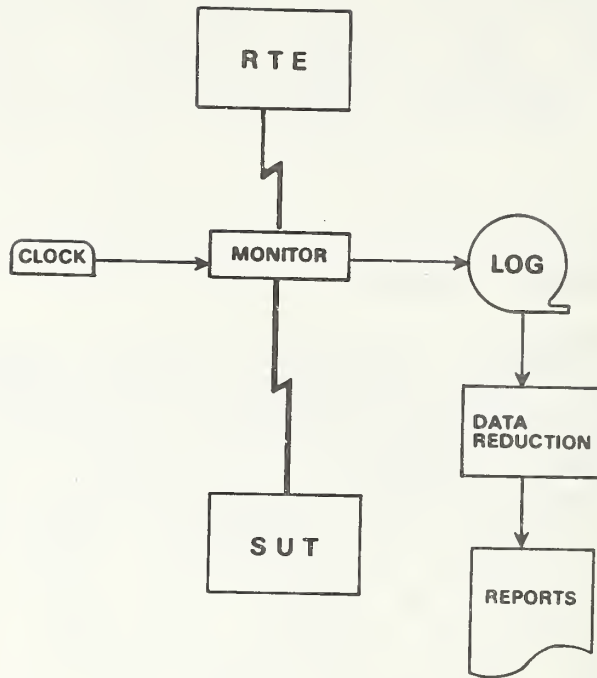


Figure 1. Remote Terminal Emulation

There are implementations of workload drivers which reside either within the SUT or in its communications front-end. Due to their location, these drivers are called internal. SUT resource dependency excludes these specific implementations from the classification RTE. In the interests of control and repeatability of testing, and of creating as near a duplication as possible of a specified workload and its effect on the SUT, the driver must be external to and independent of the SUT for the device to be an RTE.

## 2.2 Determinism of SUT's

A SUT is called deterministic if the same sequence of terminal inputs always elicits identical system responses. In this case the RTE need not provide response analysis; that is, it simply sends predetermined sequences of



characters to the SUT, expecting that the responses are also predetermined. However, in interactive usage of computer systems it is not uncommon to receive operator messages, such as, "WAIT" or "STORAGE LOW, PLEASE DELETE UNNECESSARY FILES." These unsolicited messages are random and require that an RTE be able to react to the unexpected. Teleprocessing systems by virtue of their interactive nature are non-deterministic, and unsolicited messages of varying content are a reality.

SUT determinism directly impacts the issues of repeatability and flexibility. Repeatability requires that each time an RTE presents an activity to the SUT the observed performance differences are due to the SUT and not to the RTE. Flexibility requires that an RTE not abort an otherwise useful activity due to an extraneous perturbation in the environment. There are several trade-offs in dealing with these issues depending upon the type of application. As will be discussed in section 5, in-house testing of a SUT generally requires more flexibility than does testing to assist in selecting from among alternate systems or services for the purpose of procurement. In procurements repeatability is the overwhelming concern since an unexpected response may indicate to the vendor that the SUT should be reconfigured before continuing the test.

### 2.3 Scenario

A scenario describes the user workload in a machine independent form. Functional activities to be emulated, such as the exercise of various subsystems (compilers, editors, application packages, etc.) are specified in the scenario. All actions, pauses, and decisions to be made by the emulated users are designated. Ideally scenarios are written independent of any SUT or RTE.

As an example of the importance of machine independency of the scenario, consider the use of RTE's for comparing the performance of different teleprocessing systems for the purpose of selection. The scenario insures that the workload specification is written independent of any system or special functions of a system. This implies that approximately the same effort is required of each vendor to translate the scenario into the representation suitable for the vendor RTE. This same machine-independency principle insures that the RTE's of the various vendors are also equitably treated. Figure 2 provides a sample scenario.

Enter program "A."  
Submit program for compilation.  
Correct errors in lines 10 and 15.  
Submit program for compilation.  
Correct all remaining errors.  
Enter data "B" into file system.  
Execute program "A" using data "B."

Figure 2. Sample Scenario

## 2.4 Script

A scenario is translated into a script dependent on both the SUT and RTE. The script contains the characters which constitute the user/system interaction, and the time sequence information describing the relationship among the characters. The script contains many elements in addition to a source language program and a set of data. For example, an interactive script might specify logging onto a system, creating a program to be stored in the file system, compiling, linking and loading, and executing. In execution, input and output might be conducted between the program and the terminal as well as between the program and various files. A script may contain a mixture of commands at the executive command level, subcommands and other interactions with various subsystems, programs, and data. Embedded in the sequence of this task execution are typically commands to the RTE which indicate the speed to transmit characters from the terminal device and the elapsed time to wait after receiving a system message before issuing the next input from the emulated device. Figure 3 is an overview of the scenario to script process and Figure 4 shows the hypothetical translation of a scenario statement into the text which would be included in the scripts for three different SUT's.



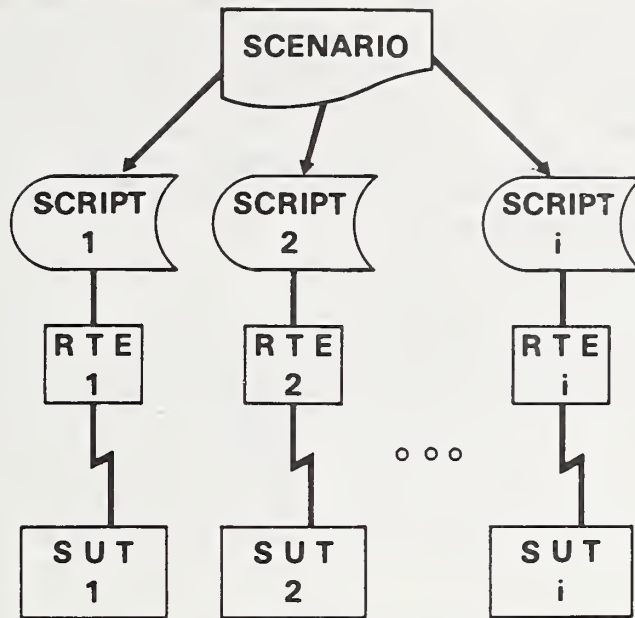


Figure 3. Scenario to Script Process

Scenario: Enter program "A."

Script 1: @ ED,I AYE  
(type text of program "A")  
@EOF

Script 2: MAKE AYE.FOR  
I (type text of program "A")  
<ESC>EX<ESC><ESC>

Script 3: TECO  
I (type text of program "A")  
<ESC>  
;U<ESC>  
AYE.FOR<CR><CR>

Figure 4. Scenario to Script Translation

A script may be thought of as the program which an RTE executes in order to perform the functions designated in the scenario. One script is required for the control of each emulated device. It is possible to permit several devices to execute the same script. Depending upon the RTE implementation, this is performed either by reentrant (shared) programs or creation of identical script files, one for each emulated device.

Dependent upon implementation, a script may be executed interpretatively or be compiled to produce machine code. In any case, it is possible for the term "script" to imply code that is human-readable or machine-readable or both. In this report script will be used to denote the human-readable representation.

## 2.5 Scene

When an RTE is applied to a SUT, the totality of characteristics associated with that application is called a scene. In order to determine the performance of the SUT, it is necessary to record pre-defined aspects of the RTE/SUT interaction. This procedure is called scene monitoring. The scene characteristics to be logged are determined by the features of SUT performance to be evaluated. An example of the type of information logged is SUT response time. The function of scene monitoring may be performed by either the RTE or a monitor independent of both the RTE and SUT.

## 2.6 Integrity confirmation

Applying an RTE to a system for the purpose of performance determination raises three questions:

1. What is the performance of the SUT?
2. Can the RTE execute the necessary functions?
3. Is the RTE performing properly during its application to the SUT?

Integrity confirmation encompasses these three concerns.

Determining the performance of a SUT by any technique is termed performance validation. Whatever motivates performance determination activities, be it system improvement or selection, it is how a system performs that is of interest.

To insure the integrity of an application of a performance validation device, the performance of that device should be tested prior to and during its application for the purpose of testing a system. Certification is the process of testing a device prior to its application. Certification corroborates that the device is capable of performing the specified functions. Consider the following example specific to RTE's; if 300 terminals are required by the scenario, certainly an RTE which only emulates 20 is unsuitable.

Certification alone however is not sufficient. Verification is the process of determining the integrity of the application of the device. Examples of the type of concern addressed in this process are; is the device executing the tasks specified in the scenario, is the device emulating the correct number and kinds of devices specified at the correct transmission speeds, and is the device recording the scene accurately for later analysis?

To summarize, the three integrity confirmation concerns are: the SUT's performance (performance validation), the ability of an RTE to perform specified functions (certification), and the RTE's performance during application (verification).

### 3. Applicability

By far the majority of RTE's were developed for in-house testing of computer systems including program development, package testing, and operations testing. In fact, one RTE was originally designed to test the design of a new line of terminals rather than a computer system.

In program development RTE's have been used for the functional testing of individual modules. Later, the various combinations of the modules are tested. Once the software is integrated, it is stress tested by an RTE.

An RTE is also useful in the area of package testing. Functional and stress testing of an entire package, performance prediction and testing, and measurement of the impact of the package on the total system are examples of this type of application.

In the area of operations testing, an RTE can aid in the determination of control limits, stress testing, and system tuning. In addition, RTE's have been used for testing the effects of hardware and system software changes, and the effects of changes in program packages. Only after RTE's were introduced for these applications, were they considered for procurement and contract monitoring.

One RTE (section 4.1) was developed for the express purpose of procurement usage (1). The RTE exerts a repeatable, controlled workload on a computer system which enables the interactive benchmarking and testing of teleprocessing systems. For procurement of computer services, RTE's also offer assistance in the area of contract monitoring. Once teleprocessing computer service is procured, an RTE can be used for periodic testing to insure that the contractually-agreed upon level of service is delivered.

---

(1) Federal Government agencies should note that at the present time the use of RTE's may not be made a mandatory requirement in an RFP nor may the results of its application be the sole basis for a selection decision. (GSA[1975]) Since changes are anticipated in Federal procurement regulations, agencies are advised to determine what restrictions may exist at the time they consider the use of RTE's.



#### 4. Implementations

There are twelve currently implemented RTE's reported in this survey. All RTE's for which the authors were able to obtain information are included. The RTE's come in varying sizes with varying capabilities. Appendices A through F summarize a few features of the different implementations.

The RTE's are presented in alphabetical order by implementor. Their descriptions are subdivided into eight outline categories. In the interests of uniformity, each RTE description is fitted as closely as possible to the outline. If there is no available information for a given outline entry, the entry has been listed even though unfilled.

Information pertaining to special or unique features of an RTE may be found in Subsection 1, "General description." Also included in this section are any psychophysical activity that the RTE performs (e.g. user think time and typing rate).

Subsection 2, "Hardware considerations," discusses both the hardware on which the RTE is implemented and the hardware which it tests: i.e., the System Under Test (SUT). Information on the RTE/SUT interface and devices which the RTE emulates is also located here.

Subsection 3, "Script preparation", contains information descriptive of the process of creating the scripts which the RTE executes.

Subsections 4 and 5 deal with the data descriptive of the RTE/SUT interaction. "Data acquisition" describes monitoring techniques, data format, and time-stamping. "Data summarization" describes the reports which are generated and the types of information summarized.

Subsection 6, "Integrity confirmation", describes any procedures that can be used in the certification, verification and validation processes described in section 2.6.

Subsection 7, "Current usage," contains the current status of the RTE in terms of its availability to those outside the organization which implemented it, and in terms of the ways in which the implementors are using it.

Potential problems and benefits are contained in Subsection 8, "Operational considerations." In addition, interesting experiments involving the RTE are described in this section.



The descriptions of the RTE's vary in length. This is strictly due to the amount of published material which is available on each device. In addition, some of the RTE's have been used in environments where publication of results is less restricted than others. For example, when an RTE has been used in procurements, this type of data is not readily available for distribution. Tables 1 and 2 preview the RTE's to be discussed. Table 1 lists the hardware on which the RTE's are implemented, and Table 2 lists the systems which have been tested by the RTE's.

NOTE: The identification of certain commercial equipment and products in this report is done to adequately identify the RTE's discussed. In no sense does the identification imply recommendation or endorsement by the National Bureau of Standards.

Table 1

## RTE Implementation Hardware

RTE	Hardware
Air Force/MITRE DVM	Data General Nova 800
Burroughs System/DCEM	B 6700 or B 7700
Control Data Corp. BARTER	Peripheral Processing Unit of a CDC system
Digital Equipment Corp. Script Machine	PDP 11/20 *
Hewlett Packard TEPE	HP 2100
Honeywell CUESTA	H 6000 or Level 66 Series 60
Honeywell DATUS	Datanet 30
IBM DB/DC Driver	IBM 370/145 *
IBM TPNS	IBM 370/145 *
Tymshare MSUS	XDS 940
Univac CNE	Univac 1100 series
Univac CS1100	Univac 1100 series

\* Minimal hardware on which the RTE may be configured.

Table 2  
Systems Tested \*

RTE	Systems Tested
Air Force/MITRE DVM	Burroughs 6700 CDC 6600 Honeywell 635 and 6180 IBM 370/155 Univac 1108
Burroughs System/DCEM	B6600 and B6700
Control Data Corp. BARTER	Cyber series
Digital Equipment Corp.	PDP 11/70
Hewlett Packard TEPE	HP 2000 and HP 3000
Honeywell CUESTA	H 6000 series
Honeywell DATUS	H 6000 series
IBM DB/DC Driver	IBM 360 and 370 series
IBM TPNS	IBM 360 and 370 series
Tymshare MSUS	Tymshare network service
Univac CNE	Univac 1100 series
Univac CS1100	Univac 1100 series

\* While the design of a specific RTE may not limit its application to these systems, this table only lists systems which are known to have been tested by the given RTE.

#### 4.1 Air Force/MITRE's Remote Terminal Emulator

The MITRE Corporation under the sponsorship of the Air Force Directorate of Automatic Data Processing Equipment Selection, the Electronics Systems Division of the Air Force Systems Command, and the Deputy for Command and Management Systems, developed two design verification models (DVM's) of their RTE in 1972 and 1973. While the majority of RTE's have been developed to test one specific system or family of systems, this emulator was designed as a general procurement tool which would be applicable across system boundaries. Therefore, unlike other RTE's which may require even a slight modification to the System Under Test (SUT), for the Air Force/MITRE RTE this approach was not acceptable. There is a wealth of documentation on this emulator and its design criteria and specifications. (See MITRE [1973-1975].)

One DVM, called the "fixed-site system," resided at MITRE, Bedford. The other DVM, called the "on-site system," was transported to the SUT. The primary use of the fixed-site system was to develop programs and scripts (2), and it connected to a SUT via the switched telephone network. The primary use of the on-site system was detailed test and evaluation of the SUT; as such it was representative of the type of emulator which the developers thought would be appropriate to a procurement selection procedure. The on-site system was moved to the site of the SUT and interfaced through cables directly to the SUT's communication line adapters. (Vol. 1, MITRE[1973-1975])

##### 4.1.1 General description

Dependent upon the DVM used, connection to a SUT may be accomplished either by local cabling or via the telephone network. To maximize real-time operating efficiency, scripts are pre-assembled to produce machine code for a DVM. This code is tailored to the specific terminal type and data control procedures for each emulated device. During execution scripts are transferred into core from disk, and are processed by an interpreter running under a real-time executive (Exec) designed specifically for this RTE. The Exec is an application-dependent, multi-tasking monitor which was implemented to free the interpreter from such general management tasks as I/O interrupt handling, data

-----

(2) The definitions of script and scenario which appear in the literature on this RTE are not identical to the terminology defined in section 2 of this report. In the interests of consistency, the authors will describe the DVM's in the terminology of this report.



The "equipment table" is central to the operation of the emulator. This table, which is defined by the user of the emulator, describes the emulated devices. Included in the descriptions of emulated devices is definition of the hierarchies of equipment such as terminals attached to a controller.

There is a script command to specify delay or think time for emulated devices. However, there is not a script command to regulate typing rate. An option does exist for an emulated device to transmit characters at a different rate than it receives them. Once specified, all characters are sent from the emulated device at that speed. That is, if a device is connected to a SUT via a 2400 bps communication line, all characters sent by that device to the SUT are at 2400 bps.

There are several commands which are available to the operator of the emulator during application. The operator may start, stop, and restart the actions of emulated devices. The status of an emulated device may be obtained, and the queries and responses associated with one or more devices can be monitored via the control teletype. There are also a set of commands which effect the emulator as a whole, rather than individual devices. For example, all error messages associated with emulator execution may be typed on the control teletype, or the data to be written to the log tape may be selected by type and by device. One interesting operator command is SCALE which allows the action of an emulator to be speeded up or slowed down.

#### 4.1.2 Hardware considerations

The Air Force/MITRE emulator is implemented on a Data General NOVA 800 with 32K words of core memory and 256K words of disk storage. The basic hardware configuration also includes: a teletype which is used for emulator control during a test, a paper tape reader which is used to enter hardware diagnostics, a magnetic tape unit which is used for off-line file storage for scripts and programs and for logging data during a test, a printer which is used to print test data from the log tape, and a Readable Real-Time Clock, an Interval Timer, and a Real Time Clock which are used for task scheduling and timing. All preprocessing and postprocessing non-real-time functions are performed under control of the Data General Disk Operating System.



The line adapters are character-oriented and permit the interfacing of up to 64 asynchronous and 8 synchronous lines. All the asynchronous line adapters operate independently with respect to transmission rate, character length, and start/stop bits. Asynchronous transmission is supported for 110 bps, 134.5 bps, and multiples of 150 bps up to 19.2 Kbps. Synchronous line adapters function independently also. Synchronous transmission rates from 2.4 Kbps to 153.6 Kbps are supported.

A large variety of terminal devices may be emulated by the DVM's. Any teletype-compatible terminal which communicates at one of the speeds specified above is supported, in addition to the IBM 2741. However, the only asynchronous terminals which have been emulated in a testing environment are teletype-compatible at 110 and 300 bps and IBM's 2741 at 134.5 bps. At this point the synchronous terminals emulated in tests have been the IBM 2780 remote-batch terminal and the Burroughs remote-batch terminal.

A basic design concept was a totally SUT independent emulator. A logical point of connection for such a device is the EIA RS-232 interface.

The fixed-site emulator has been used in tests of the following systems: IBM 370/155, CDC 6600, Burroughs 6700, Honeywell H635 and Univac 1108. In addition to testing the same types of systems as the fixed-site emulator, the on-site emulator has tested Honeywell systems 6180 and 635 under GCOS and Multics.

#### 4.1.3 Script preparation

Scripts exist as disk files under the Data General Disk Operating System. The Data General Editor is the principal tool for creating and changing scripts. Interaction with the Editor is conducted through the control teletype for the emulator. As an aide to the script writer, a "Macro Processor" has been implemented. The user supplies a master directory of macro names, and then submits a script for processing. As the input script file is scanned, the appropriate expanded text is written to the output file whenever a macro name and its associated arguments are encountered. The output file contains text obtained from either the input file, the body of a macro, or a macro argument.

After the script has been processed by the Macro Processor, it is "assembled." Scripts created by emulator users consist of ASCII character strings written with such user aides as field separators and labels. Even after the

Macro Processor has been used, the script is still in a form which is designed for expedient human, not machine, use. To facilitate real-time operation, the assembler creates a file which is tailored for emulator interpretation and execution. Scripts are created which reflect the specific terminal type to be emulated and the data communications control procedure to be used.

A number of conversion tables and procedures are used to implement this translation. For example, prior to assembly, a script written to emulate a 2741 terminal specifies the character strings to be sent to the SUT in ASCII; after assembly, it contains the actual EBCDIC code to be sent. Also included in the assembly process are the insertion of the start of message and end of message code, and certain error checking functions. It is the code created by the assembler which is executed by the emulator.

#### 4.1.4 Data acquisition

The Readable Real-Time Clock is used for time-tagging events logged on a 9-channel magnetic tape. Two timetags are associated with every user and system message: the times of the first and last characters. After a test is completed, the data on the magnetic tape is processed to determine response time characteristics.

#### 4.1.5 Data summarization

The emulated terminal input is viewed as the impetus to which the SUT reacts; therefore, a user character string is called a query and a SUT character string is called a response. There may be a character or group of characters, occurring at the beginning of the characters sent by the SUT in response to a query, which are sent simply to acknowledge receipt of the query and are not part of the true response. Suyemoto[1975] notes that these characters preceding the true response formulate an acknowledgement, and are generally sent almost immediately upon receipt of a query; thus a system which may have a very sluggish true response time may appear to render almost immediate response service. Because the DVM's log only the first and last characters sent by SUT, difficulty arises in attempting to interpret what the true response time is when a system sends acknowledgements.

The DVM's fill response buffers until one of the following occurs:

1. a buffer is filled to a previously defined size;
2. an end of message is received; or
3. a timeout (a previously defined maximum response inter-character arrival time) occurs.



It is a responsibility of the data reduction program to determine if an acknowledgement exists and what the start of the true response is. Obviously, if the SUT sends no acknowledgement the two time tags logged will accurately reflect the SUT's performance. When a timeout occurs, SUT characters are directed to a different buffer and again the time of the first character in this buffer is logged. So if the acknowledgement is isolated in a buffer, it can be ignored; and if a true response is contained in a separate buffer, the data reduction program can isolate this buffer and use only the time tags associated with it. However if the true response is contained in the same buffer as the acknowledgement, no determination may be made as to the start of the true response except that no SUT inter-character arrival times exceeded the timeout.

The predominant measure is response time. While logging the number of SUT characters provides a measure of system verbosity, the data reduction program, which will be described later, summarizes performance in terms of response time.

The data reduction program, DATAR, generates three types of reports which may be printed on a line printer or on the control teletype. The general options include: a brief summary of the entire test, detailed summaries produced separately for each device, and detailed listings produced either jointly for all devices, or separately for each device. There are many options available for detailed summaries; for in-depth descriptions of not only the reports, but also DATAR software, see Volume 7 of MITRE[1973-1975].

An example of the use of DATAR is a user requesting a brief summary of a test. Upon reviewing the summary printed, the DATAR user sees that the maximum response time is unacceptable. To investigate, the user may request a detailed summary of the device causing the maximum response time. This summary may indicate the reason for the deviation from the average for that test.

#### 4.1.6 Integrity confirmation

There are several features of the DVM's which lend themselves to use as integrity confirmation techniques. The operator command to randomly select a given device to monitor during a test provides a means to insure that devices are executing the correct scripts. DATAR's brief summary provides two measures of the emulator's performance: percent of total emulator CPU time used and total CPU time. These two measures provide an indication of how the DVM is reacting to the workload it has to emulate. Certainly as

per cent of idle time decreases and per cent of CPU increases, the operator can discern if the saturation point for the emulator is approached.

In addition to the diagnostic routines available through Data General and the Disk Operating System MITRE implemented a set of diagnostics and error message routines. The diagnostics provide a means for testing the modem connected to the communications interface, the hardware for the Readable Real-Time Clock, and the circuit logic for the Interval Timer. Volume 9 of MITRE[1973-1975] discusses the routines available and a variety of troubleshooting techniques. Typically diagnostics could be run prior to every testing situation to insure that the emulator is functioning properly. The DVM's log error conditions during application and the data reduction indicates the error messages occurring on each device. This facility is another form of insurance that the emulator is functioning correctly during a test.

#### 4.1.7 Current usage

The DVM called the on-site system is currently at the Data Services Center at the Pentagon. It is being used to test two computer systems: a Honeywell 635 running GCOS, and a Honeywell 6180 running Multics. The fixed-site system is located at the MITRE Corporation in Bedford, Massachusetts.

#### 4.1.8 Operational considerations

The Air Force/MITRE emulator has been rigorously tested on different systems, and the results documented. In this section two different experiments will be described.

Peterson[1974] discusses the application of the emulator to five different SUT's. Three common scenarios, representative of a user workload across system boundaries, were developed: EDIT in which the editing capability of a SUT was utilized, BASIC in which short, computational problem-solving was performed, and FORTRAN cost in which a simple accounting program accumulated and printed costs.

The major problems involved scenario and script development. Translating scripts from one SUT to another was difficult due to the poorly defined and documented data communications procedures. Line control procedures varied from facility to facility even for equipment of the same manufacturer. Differences were found between documented procedures and actual operation. In fact procedures vary from day to day due to changes in system software.



However, it was determined in these experiments that it is feasible to develop equivalent on-line workloads if the scripts are limited to commonly used system commands and standard programming languages.

Crothers[1976] describes application of this RTE to an IBM 370/155 and IBM 370/158. Sixteen IBM 2741 terminals were emulated doing heavy editing, data entry, and FORTRAN execution at the same time as a standard batch background job stream was run simultaneously. The results of this experiment were valuable in the identification of specific problems recorded in each emulator run. For example, failure to send line numbers during heavy use of the TSO editor was a problem. For each line of data to be entered, TSO first provided a line number. During heavy emulated use, some of the line numbers were replaced by synchronization characters. This application illustrates that the emulator can make contributions as a tool for testing new releases of software.

## 4.2 Burroughs' Data Communications Traffic Emulator (System/DCEM)

### 4.2.1 General description

The Burroughs Data Communications Traffic Emulator, formally abbreviated "System/DCEM" (3) and usually referred to as "the emulator," is used to emulate a specified data communications load in accordance with user specifications.

The traffic emulation is controlled by the contents of script files, and by system commands entered from a control terminal or the system console. A large number of system commands are available for control over the emulation.

There is a script command to specify think time (delaying an interval of time after receiving a message from the SUT before sending the next emulated user message). Delay time may be specified in place of, or in addition to, think time. Delay time is defined as the interval of time from the end of one emulated user message to the beginning of the next user message. It is possible that if the SUT does not respond as quickly as expected, that the interaction could get out of synchronization if only delay time were used. Characters sent from an emulated device to

-----

(3) This program package is subject to proprietary protection.

a SUT are transmitted with no inter-character delay time; that is, the full capacity of the transmission line is employed.

#### 4.2.2 Hardware considerations

The emulator can run on any Burroughs large system - B6700 or B7700. A subroutine, called "Request Set" in Burroughs terminology, runs in the front-end processor as part of the emulation. The Emulation Request Set performs the communications functions necessary to provide the expected terminal reactions to situations produced by interaction with the System Under Test.

The emulator is designed to handle any number of terminals and any data communications configuration which can be managed by the data communications subsystem in the SUT. The terminals which can be emulated include buffered CRT terminals operated in a polled environment using synchronous and asynchronous line disciplines, remote batch terminals, and teletypewriters. The teletypewriters can cause problems with the half-duplex line control, a problem Burroughs calls "contention" because there is not a slave-master relationship between RTE and SUT. Due to their procedure for calculating the time at which to send a user message, it is possible for the emulated teletypewriter and SUT to attempt to transmit messages simultaneously. In half-duplex communications attempts to transmit in two directions at the same time are disastrous. There is an optional procedure designed to eliminate this problem. This procedure consists of a modification to the front-end program in both the emulator and SUT which checks whether there is ongoing communication before initiating transmission.

The emulator is created by an "emulation generator" which builds a description file according to the information supplied by the user concerning the number and names of emulated terminals, the files to be used, and a set of options. If requested, the generator will have the version of the emulator specified by this input compiled.

The emulator has the capability of running internally within one machine. In this case the emulator, along with special patches to the communications software, causes messages to be placed on the target queues without using any physical data communications lines.

Input for the emulation generator is expected to be in card format. There are three types of input records: emulation, file, and optional statements. The emulation statement gives the name of the emulation and an optional



identifier to be printed at the control terminal. The file statement names a script file and gives a list of all emulated terminals which will use that script. Many terminals can employ one script, or an individual script can be used for each terminal. Among the optional control statements are a statement to cause compilation of the emulator, and a statement to build the internal format script files from the card image files.

System commands are used to control the operation of the emulator. Depending on how the emulator was built, they may be entered from a controlling terminal or the system console. The following are among the commands available: an abort command to terminate the emulation; a compensate command to allow the emulator to delay less than the time specified in the script to compensate if the actual delay ever exceeds the requested delay; a command to scale the interpretation of the delay specified in the script; commands to start and stop an emulated terminal; a verify command which causes the emulator to verify all messages sent; information and status requests which interrogate the operational condition of the emulator; plus a set of debugging commands.

#### 4.2.3 Script preparation

A script is used to control the communications traffic from each (emulated) terminal. The symbolic input script file contains one message to be transmitted from the emulated terminal to the SUT per record. Records may be of any length, but the file must have fixed length records.

If there is a special character in the first character position of the record, the record has a special function. These functions include specification of an expected response from the SUT which is to be checked and specification of the delay to be taken between messages. The output file built by the generator contains all the information from the input file, plus control information. In operation, the emulator first reads and acts on the control information, then transmits the stimulus message.

This script consists of individual messages set up like an actual session from a live terminal. Each message is sent after waiting either a specific delay interval or until a specified response is received at that terminal. The delay used, if any, may be changed for each message or may be a random number within a given range.

Since the execution of the emulator can be controlled from a selected terminal, the starting of the scripts may be controlled to occur all at one time or staggered.

If non-standard or complicated actions are to be dependent on the data received, the emulator can be generated with links to external procedures. These external procedures can provide file access, or can include part of a response in the next stimulus to be sent to the SUT.

#### 4.2.4 Data acquisition

During operation of the emulator, all input and output between the emulator and SUT is recorded. There are two different procedures for timing the interaction between the RTE and SUT, either of which may be selected. The first method of timing a message is unique among the RTE's studied. A message from the emulator to the recipient SUT is time stamped upon receipt of an ACK from that SUT; this is the time at which the communications software knows that the message has arrived successfully and can be safely discarded. Similarly, messages from the SUT are time stamped when successfully received and an ACK is transmitted back to the SUT. The second method of timing provides an alternate time stamp for the message sent from the emulator to the SUT; the time stamp on the message from SUT to emulator is unchanged. The alternative time stamp on the message from emulator to SUT occurs when that message is queued for output within the emulator.

#### 4.2.5 Data summarization

The tape file can be read by an existing analysis program which has the ability to print the entire tape, or to extract the communications for a specified terminal. Further analysis programs must be specially written for each benchmark demonstration. In this way a library is being created.

#### 4.2.6 Integrity confirmation

A live terminal can observe the data traffic for any emulated terminal. This live terminal, called a "line monitor," can be switched from line to line.

A live terminal with operator can be connected to the SUT through the RTE and have the same data collected as that collected for an emulated terminal.



#### 4.2.7 Current usage

The emulator is presently employed only as a tool in marketing support. It might be available upon request on some price quotation basis.

#### 4.2.8 Operational considerations

The optional script command "compensate", which was described above, permits an interesting adjustment feature during operation. If the actual delay time should be excessive, succeeding delay times can be reduced to compensate.

There is currently no provision for transmission from the SUT to the RTE for batch terminals. The design assumption is that files are created at the SUT and/or printed there.

### 4.3 Control Data Corporation's Project BARTER

#### 4.3.1 General description

The Control Data Corporation RTE described here is the result of the BARTER (Benchmarking Approach to RTE Requirement) project being conducted at the Benchmarking facility for use in meeting RTE benchmark requirements. BARTER is the combination of two previously existing CDC products. One of them, "the timesharing stimulator," is an adaptation of a standard product available in their multiprocessing system. The principal modification of this version is that it is external to the System Under Test (SUT). The other major component of BARTER is "the external Remote Job Entry (RJE) load tester."

The psychophysical phenomena emulated for interactive users include think time, which may have a constant plus a random component, and typing rate.

#### 4.3.2 Hardware considerations

The description of the operation of the RJE load tester and the stimulator is dependent on a very brief discussion of the architecture of Control Data products. For purposes of this discussion, it is important to note that each CDC system consists of one central processor unit (CPU) and up to 20 peripheral processing units (PPU). Each PPU is a processor capable of executing instructions and comes

equipped with its own memory. It also shares memory with the CPU.

The external RJE load tester is a program which utilizes one peripheral processing unit to emulate the data traffic from a Control Data remote batch station. The communications lines discipline is synchronous employing 8-bit characters with half-duplex line utilization and block transmission. Each PPU can emulate 16 such terminals operating at 4800 bits per second or 8 terminals at 9600 bits per second. Terminal requirements in excess of this can be accommodated by running another copy of the external RJE load tester in another PPU.

The timesharing stimulator runs using one peripheral processing unit to emulate up to 64 asynchronous lines. Each line is teletype-compatible (units 33/35), employing a 64 character ASCII subset. The capability for employing the full ASCII character set exists in the stimulator code. The interaction of the stimulator with the SUT is dependent upon the SUT terminating output with a specific nonprinting character which signals the end of system output and a willingness to accept input from the user.

The stimulator is initiated from the system console of the system supporting the RTE. Among the parameters which may be specified from the system console are the number of terminals to emulate, the line speed and characters per second, the input or typing speed characters per second, the think time delay in seconds, the activation count (the number of terminals to activate every specified number of seconds), the number of times to repeat the stimulation, and whether data logging is requested. In addition to these specifications, it is possible to assign specific scripts to teletypes and to adjust the line speed, think time, and repeat count by the assignment.

#### 4.3.3 Script preparation

The operation of the external RJE load tester is controlled from a deck of cards. Typically, the batch workload is presented as actual jobs, a replication count, and the time limit in which the jobs are to be accomplished. The assignment of a job to an RJE terminal is specified.

For the stimulator, each script to be emulated is created as card image file. This file contains all of the terminal input to be emulated plus a few additional stimulator specific parameters.

#### 4.3.4 Data acquisition

The RJE load tester records all output on a file. It has a utility program which, for each block of input and output, will print out the first line of that block, plus control code information and the time at which that block occurred.

The stimulator collects all data received and sent on a file. The time of the carriage return terminating input from the emulated user, and the time of the first character output by the SUT are logged.

#### 4.3.5 Data summarization

Post processing of the stimulator data is accomplished by two programs. The first demultiplexes the logged data for each terminal. The terminal data appears as it would on a terminal printer page. Time stamps are present for the carriage return terminating input and for the first character of output. Following demultiplexing, data may be subjected to further analysis to generate response time statistical data by a Fortran program. The output of this program is a response time profile as well as the mean, mean deviation, and standard deviation values. Data may be reduced for all the emulated terminals, or for a selected set thereof.

#### 4.3.6 Integrity confirmation

The operation of the RJE load tester may be validated by running the same emulated job stream through a real terminal. The log tape of the SUT is also available for validation.

The operation of the stimulator may also be verified by means of the log tape. In addition, the communications to and from any emulated terminal may be output on a real terminal attached to the RTE system.

#### 4.3.7 Current usage

BARTER is being implemented by internal development personnel for checkout of their products as an internal tool. It is available and in general use by CDC in their benchmarking facility, but not available for use outside the benchmarking facility.



#### 4.3.8 Operational considerations

The stimulator does not check the output from the SUT. In the case of SUT malfunction, it is therefore possible for the stimulator and SUT to get out of step.

#### 4.4 Digital Equipment Corporation's Script Machine

Digital Equipment Corporation (DEC) has developed several different drivers. SCRIPT, an internal driver for the DECSys-10, simulates some, but not all, of the interactions between a user terminal and the System Under Test (SUT). In particular, it does not simulate the interrupts received from the communication hardware to the CPU.

SIMLOD was developed to simulate active terminals on a DECSys-10. SIMLOD runs in the PDP-11 which is part of the DC76 front-end communications module. Even though SIMLOD is implemented external to the CPU of the DECSys-10, it is internal to the entire system under test. Because it is resident in the communications front-end, it may not be classified as an RTE as defined in section 2.

An RTE has been implemented at DEC which runs on a PDP-11. This RTE is called "the script machine" and is discussed in Turner and Levy[1976] and Kosko and Turner[1975]. DEC's script machine will be described in this section.

##### 4.4.1 General description

The script machine evolved out of the need to test in a repeatable manner a new PDP-11 operating system called the Interactive Application System (IAS). Although the original design impetus involved IAS, the script machine is not bound to this operating system; in fact, the script machine may be used for various applications involving the PDP-11/70 as will be described. The script machine is completely external to the PDP-11/70 being tested and connects via communication lines.

Specified in script files are the characters to be sent to the System Under Test (SUT) by the emulated terminals and user think times. User think time is defined by Turner and Levy[1976] as the elapsed time between the completion of processing one request by the SUT and the first character typed by the user in the next request. Included in this specification of think time are two components.



Non-overlapped think time requires that think time not begin until the completion of the SUT's processing of the preceding command, while overlapped think time can overlap the processing of the preceding command. Another interesting feature is that think times may be explicitly assigned values or may be obtained by a discrete probability distribution which is included in the script. Characters sent from an emulated terminal to a SUT are transmitted with no inter-character delay time; that is, the full capacity of the transmission line is employed.

The script machine has been used to drive the 11/70 under two different operating systems: IAS in support of interactive and computational users in a conversational mode, and RSTE/E in support of a data management application package. Both of these operating systems support full-duplex communications. In full-duplex communications the terminal and system may send characters simultaneously, and the user can type-ahead or continuously send commands without waiting for the system's response to individual commands. Neither of these operating systems normally sends a herald character at the completion of processing, such as a keyboard unlock; therefore, there is no uniform way for the script machine to recognize that the SUT has completed its processing. While an "expected response" command exists, it is only used to ensure that the SUT is responding correctly. Rather than require the use of this command throughout scripts, the operating systems tested are modified to send a special "prompt character" each time a user's request has been processed. The time at which this prompt character is received is central to the operation of the script machine. Based on this time, response time to the preceding command and the start time for non-overlapped think time for the next command are determined.

#### 4.4.2 Hardware considerations

The script machine is currently implemented on a PDP-11/20 which connects to the PDP-11/70 via 32 serial lines with EIA interface. A line from one system is simply connected to a corresponding line of the other system. In this process all characters written to a serial line by the script machine are taken as user input to the 11/70; and, in reverse, all characters written to a terminal by the 11/70 are read as input to the terminal by the script machine. The option exists to record on a log file on the disk of the script machine all characters sent by the 11/70 to a selected set of lines.

#### 4.4.3 Script preparation

Scripts reside on ASCII disk files. They may be prepared with any text editor which runs under the RT-11 operating system.

#### 4.4.4 Data acquisition

All characters written to a terminal by the SUT may be recorded in a log file on the script machine disk. However, the script software has the capability of evaluating the performance of the SUT as the test is run. To use this capability, the script writer specifies the expected response and a minimum response time. Each time the script machine receives an expected response, it checks if it matches the expected response. If the expected response is not received, an error message is typed, and the operator may abort the run if desired. If the actual response time is significantly less than the minimum specified, the run, again, may be aborted. The script machine computes the ratio of the actual response times to the specified minimums, and maintains a frequency distribution of these ratios.

The script machine also monitors its own performance during a test. A record of its promptness in sending out user commands and of the backlog of lines of input test to be processed is maintained. Determination of script machine saturation is possible by computing these values.

#### 4.4.5 Data summarization

The data summary routines for the script machine appear to be very flexible. In the IAS experiment the key concern is the determination of the conditions under which a given configuration provides satisfactory response. This approach includes characterizing each hardware configuration of IAS in terms of users supported within given response time criteria. These criteria are:

- a. that 90% of the responses should be less than the larger of i) 5 times the stand alone response time and ii) 1 second; and
- b. that the sum of the actual response times should be no more than 5 times the sum of stand alone times.

Two types of users are defined for IAS: interactive who edit text and do simple file manipulation, and computational who compile, link, and execute compute bound FORTRAN programs. Various combinations of these two user types are used to determine if response criteria can be met under the given loading.

In the data base management experiment, the measurables are the total number of transactions, the number of disk accesses, the per cent of system utilization by user, I/O, Executive and idle, the total number of characters sent and received by the terminals and SUT, the number of characters sent per second per line, and the number of characters sent per transaction. The SUT keeps track of the disk access and system utilization data and the script machine the other data.

#### 4.4.6 Integrity confirmation

The self-monitoring function of the script machine is an expression of concern for RTE verification. It would be interesting to know what, if any, effect this extra function has on the script machine.

The option to log all characters sent by the SUT to the script machine is a SUT validation procedure which insures that the SUT is executing the specified scripts.

#### 4.4.7 Current usage

##### 4.4.7.1 The IAS experiment

In this experiment each test consisted of a minimum of 15 minutes of SUT operation. Scripts were started at 5 second intervals to prevent lock-step execution in which every script is issuing the same command simultaneously. The scripts were repeated continuously until expiration of a time limit; at that time, the response data was printed, the configuration rated on a pass/fail basis. Basic to the determination of the boundary for satisfactory response regions is the premise that if a system meets response criteria at a specific point, it can also meet them at any point below that loading point; and if it fails at a specific point, it will fail at points above this.

##### 4.4.7.2 The data base management experiment

Two different types of scripts are used in this experiment: one to log the 27 Data Transaction Programs (DTP's) onto the PDP 11/70 RSTS/E system, and one to get the jobs started and supply the emulated user inputs. In addition to the 27 DTP's, an RSTS/E error logging facility and a system status video display program are also logged in.



Prior to execution of the second type of script, system status and disk usage reports are obtained from RSTS/E. During execution of the scripts, RSTS/E continues to audit system status and disk usage while the script machine audits character processing by the communications equipment. At the completion of a run, the stop time is logged and again the system status and disk usage reports are obtained from RSTS/E.

The experiment described ran for 11 hours 43 minutes and 50 seconds. No errors were logged by the RSTS/E error logging facility, and by inspecting the job status and by monitoring the system during the experiment, it was determined that there were no problems with the DTP's. One-half million transactions at a rate slightly greater than 13 per second were processed during the test.

#### 4.4.8 Operational considerations

The published reports on usage of the script machine indicate that the tests have been successful. The only problem appears to be the universal one of workload determination. In the IAS experiment the satisfactory response region is determined by loading the system with the maximum number of computational users, and then adding interactive users one at a time until the system fails. At that time one computational user is eliminated and the process of adding one interactive user at a time until system failure begins again. In this way the load levels and configurations are determined.

In the data base management experiment there are nine different scripts in triplicate to produce the 27 emulated users. Each script contains five unique transactions, and each script is executed 4115 times. The data base records used for each transaction are randomly selected on the basis that records requiring more than one probe by the hashing algorithm are rejected. The hashing algorithm is based on an identification number, and facilitates record lookup. The justification for the procedure of developing the transactions used is not given.

A look to the future was provided by Turner and Levy[1976]. In addition to the dimensions of interactive versus computational usage in workload definition, they would like to add a dimension for real time load. A real time process which interrupts the SUT at a set rate and performs a minimal task at interrupt level will be defined. The measure of workload at this dimension will be the frequency of interrupts, and the performance criterion will be not to miss any interrupts. These authors see the script machine as a means to make intelligent configuration decisions.



## 4.5 Hewlett Packard's Timesharing Event Performance Evaluator (TEPE)

Hewlett Packard (HP) has implemented an RTE called Timesharing Event Performance Evaluator (TEPE). TEPE was developed by the Data Systems Section for use internal to Hewlett Packard for system development and testing. (Hawkes[1975])

### 4.5.1 General description

The script language allows the selection of one of several think-time options for each emulated terminal. These options include two random think time options (each with a different mean think time) and several constant think time options. Think times can also be set within a script to apply to only a single message transmission to the SUT, in order to handle special cases.

Typing rate is initially set for each terminal in the initialization file (see section 4.5.8 for a description of the initialization file) for the emulation, but can be changed during the actual test.

TEPE allows for extensive user interaction while a test is being run. Emulated terminals can be selectively monitored, "disconnected," or have their characteristics changed.

### 4.5.2 Hardware considerations

TEPE is implemented on a HP 2100 system under DOS III and was designed to drive a wide class of systems. At the present time, it has been used to drive both the HP 2000 and HP 3000. In addition to the HP 2000 series CPU, a magnetic tape drive is required. The connection between TEPE and the HP 3000 System Under Test (SUT) employs a multiplexer protocol. Only one cable is employed to connect 16 emulated terminals. Up to 32 asynchronous terminals can be emulated by TEPE; however, tests of an HP 3000 have been made employing two HP 2000's running TEPE which brought the total workload to 64 emulated terminals. Emulated terminals may transmit at a rate of 110 to 2400 bps.

TEPE is written in assembly language. Timing is based on a crystal controlled clock.

Like many other RTE's, TEPE requires the SUT to send a special prompt character at the end of every SUT message. However, the TEPE designers foresaw the potential difficulty in not checking the received SUT message to insure that it

is the correct or expected message. Each SUT message is, therefore, optionally compared to a predefined expected message; alternative actions are possible depending upon whether there is a match or not. Message checking can be disabled on an individual message basis.

#### 4.5.3 Script preparation

Scripts are prepared on an HP 3000 computer system using the HP 3000 editor, EDIT/3000. The script language contains special TEPE recognized commands to perform such functions as SUT response text matching or think time setting for a particular message in a script.

The scripts are "readable" in that they contain the exact text that will be transmitted to the SUT (in source form) and they allow insertion of comments which are ignored by TEPE.

#### 4.5.4 Data acquisition

All characters transmitted between emulated devices and the SUT are recorded on magnetic tape for processing after a test is complete.

#### 4.5.5 Data summarization

SUT response time is the primary measure of SUT performance for which statistics are generated, but other information such as throughput may be readily deduced.

Data reduction is performed on an HP 3000 via a FORTRAN package. Several special features are available in the reduction programs. These include: selection of response time definition, selection of the time window within the emulation log record to be processed, histogram displays of the response time distribution over the range of response times, and selection of the types of events for which statistics should be calculated.

#### 4.5.6 Integrity confirmation

Hardware and software monitors have been used to verify the internal operation of TEPE.

#### 4.5.7 Current usage

TEPE is currently only available for use internal to Hewlett Packard for system testing. The usage breaks down into two categories: performance evaluation and reliability testing. TEPE has also been used in stress testing cases and debugging system changes.

#### 4.5.8 Operational considerations

There is an overall initialization file which controls the operation of all executed scripts. The initialization defines such variables as the prompt characters (prompts may be associated with both SUT and emulated user messages), the length of the test, the number of terminals emulated, the file names of the scripts, and think times. In addition the initialization strictly defines the logon procedure for the emulated devices. The initialization file procedure loads the specified scripts and begins execution. Relative times for device logons are specified along with the maximum interval of time to wait for the SUT to process the logon. If the time interval for logon is exceeded an error message is printed on the operator's console.

Throughput, here defined as the amount of CPU time received by a user, is recommended as a good correlation measure with response time. Throughput is evaluated by internal monitors. The repeatability provided by TEPE is viewed by the developers as providing an adequately controlled environment for application of internal monitors.

### 4.6 Honeywell's DATUS

In 1968 Honeywell designed an RTE, DATUS, as a testing tool for the interactive loading of computer systems. In the spring of 1969 use of DATUS began, as did its evolution from strictly a test tool to a benchmark and measurement tool.

#### 4.6.1 General description

DATUS was designed to be completely external to the System Under Test (SUT) and to provide any type of teletypewriter terminal environment for loading, measuring and benchmarking tests.

An interesting option of DATUS is the probabilistic selection of emulated user commands. It was noted by the designers that when the same scripts were executed on many



emulated terminals, the execution of script commands began to cluster around the execution of a single command which required substantial CPU time. To avoid lock-step execution of scripts, the option was implemented to execute emulated user commands based on a probability distribution. For example, given emulated user commands A, B, and C, each command is assigned two parameters: one is the per cent of the emulated user session in which the command should be issued, and the other is the probability of each command following the others. The commands are selected in real-time, and thus the scripts are dynamic. The option to specify a fixed sequence of commands in a script is also allowed.

Emulated user think time may be specified explicitly by a script command or on a random basis via a table. There is a "manual mode" which presents a worst-case test with all lines simultaneously executing the same script. There is not a script command to control the typing rate between emulated device and SUT; rather characters are sent at the full speed of the transmission line.

#### 4.6.2 Hardware considerations

Implemented on a Datanet 30, this RTE connects to the SUT through data set eliminator cables. The software consists of an operating system, script interpreter and data reduction program.

The hardware configuration includes the Datanet 30, a control terminal, a monitoring terminal and a magnetic tape drive for data logging. The maximum number of devices that DATUS can emulate is 52 using 110 bits per second transmission lines, 37 using 150 bits per second lines, or 17 at 300 bits per second lines.

DATUS is used for testing large Honeywell computer systems of the 6000 series. In the interests of conserving the RTE storage and processing time needed for the context scanning of each system message to determine if the system message is complete, it was decided to require the operating system of the SUT to send a special prompt character when it had completed a message. This special character, which may be an Xon (DC3) or some other specified character or sequence of characters, is an indication to DATUS that a SUT message is complete and that processing may continue. While this procedure simplifies the "end of message" search, when system messages are not scanned for content, it is possible for the RTE and SUT to get out of synchronization. This occurs when the SUT sends an "unexpected" message and the RTE interprets it as the correct message by virtue of it receiving an Xon. DATUS has the capability to check

expected responses to any number of characters, but this procedure requires additional core storage.

#### 4.6.3 Script preparation

Scripts are punched on cards for input to DATUS, and they are reentrant.

#### 4.6.4 Data acquisition

All channel activity between an emulated device and the SUT are logged along with the time of day and character direction (that is, was the character sent by DATUS or SUT). Due to the level of detail of the log tape the data reduction program can calculate many measures.

#### 4.6.5 Data summarization

The data reduction program calculates response and think times, prints a complete conversation record, and allows task summaries. A task is a class of user activity. For example, invoking an editor is a task. Thus the capability exists to summarize the response time associated with a specific task.

#### 4.6.6 Integrity confirmation

The complete conversation record with associated timing is the primary means used to verify RTE operation. The record for each emulated device is examined following a test to insure that the scripts were executed properly.

#### 4.6.7 Current usage

While DATUS is still being used as a testing tool at the benchmarking facility, it has generally been superseded by another RTE called CUESTA (section 4.7).

#### 4.6.8 Operational considerations

While DATUS proved an invaluable tool to the benchmarking facility, its capability is limited by the Datanet 30. The Datanet 30 limited its capability to handle high bandwidth communication and was insufficient to emulate Honeywell's video terminal VIP.

## 4.7 Honeywell's Communications User Emulated System for Traffic Analysis (CUESTA)

Honeywell developed the Communications User Emulated System for Traffic Analysis (CUESTA) to provide extended capability for computer system testing over what was available in DATUS (section 4.6).

### 4.7.1 General description

CUESTA is implemented as an extendible executive system with fundamental emulation control. The script is a module which provides the detailed emulation functions through the script-command processors. Script writers are able to add and/or modify script commands and operator control commands during the generation of scripts. Scripts control not only the operator dialog, but also contingency decisions and the major portion of the line discipline.

Like DATUS, CUESTA provides for the optional execution of script tasks based on the probabilistic occurrence of previous commands or tasks. Associated with each task is the percent of an emulated user session in which the task should be executed and/or the probability of each task's transition to following tasks.

Emulated user think time is specified either explicitly by a script command or a probability distribution may be used. Additionally, think time may be specified for each emulated user message or for an entire emulated user session. The mode of specifying think time may be altered during script execution via an operator control command.

There is not a script command to control emulated user typing rate; all characters are sent at the speed of the transmission line connecting emulated device and the System Under Test (SUT). However, script writers may incorporate typing rate into scripts by segmenting the total transmission and/or developing their own script commands.

Another similarity to DATUS is the requirement that the SUT send a "prompt" to indicate that it has completed sending a message. However, unlike DATUS, CUESTA can prompt on any number of characters for a given script command and can provide to the script writer the capability to make contingency decisions based upon which prompt was received. A prompt may consist of any number of consecutive characters embedded within, though normally at the end, of the received SUT message.



#### 4.7.2 Hardware considerations

CUESTA is implemented in a Honeywell Series 6000 or Level 66 Series 60 CPU operating under GCOS with a Datanet 355 front-end processor. Special software is required in the Datanet for the RTE. CUESTA connects to a SUT through modems and/or modem eliminator cables.

CUESTA can emulate up to six time division multiplexors each supporting 52 terminals at 110 bps or 17 terminals at 300 bps. Thus, these groups of 52 or 17 asynchronous terminals can be emulated on one synchronous time division multiplexor interface channel. In addition, CUESTA can handle 96 lines at rates from 110 to 9600 bps which may be any combination of synchronous and asynchronous, multidrop terminal devices (teletype-compatible, remote-job entry, or Honeywell's video terminal VIPS).

Although CUESTA has only been used for the test of Honeywell systems, there do not appear to be any design constraints that would limit CUESTA's application to other systems. The primary difference in such an application is that the cables between CUESTA and SUT would have to be one-for-one with respect to the emulated devices.

#### 4.7.3 Script preparation

The script is generated from ALGOL-like source statements which are processed by a special compiler. The output is a CUESTA program module. Standard script command processing routines are provided in a library. Script writers also have the capability to write their own script command processors and operator control command processors and include them with their script source statements.

Scripts are re-entrant; thus, emulated users may execute any appropriate portion(s) of the script without regard to the other emulated users.

#### 4.7.4 Data acquisition

All channel activity between an emulated device and the SUT are optionally logged along with the time of day and character direction (that is, was the character sent by CUESTA or the SUT). Script writers have the capability to have logged any information they deem desirable. Thus, the facility exists to perform a variety of analyses following a test. Data may also be collected in a set of dynamic data collection buffers specified by the script writers when they invoke the on-line data reduction facility. Standard script commands are provided to accumulate time intervals and transaction rates.

#### 4.7.5 Data summarization

Data summarization is done in one of two ways: on-line summarization and reduction of the log tape following a test.

On-line summarization is limited to the data collected by the script. The script specifies when to capture time intervals and when to print the summarization of the captured data. The output identifies the summary/collection buffer by name (which is specified by the script writer) with its respective minimum/maximum, average and standard deviations.

The reduction that is done following a test is accomplished by the use of a number of programs. These include the calculation of response and think times by individual emulated user or by groups of emulated users, a complete conversation record, and task summaries. Users have the option to write their own summarization routines.

#### 4.7.6 Integrity confirmation

The complete conversation record with associated timing provides a post-processing procedure to verify RTE operation. In the real-time there is also the ability to dynamically select an emulated device and, in turn, have all emulated device and SUT characters diverted to a line printer. The random nature of switching between emulated devices insures that all devices are executing the specified scripts at the proper time. The ability to dynamically summarize data may also be used as an integrity confirmation technique depending upon the data collection specified in the script.

#### 4.7.7 Current usage

CUESTA is available for use in the benchmarking facility of Honeywell. It is also used as an engineering tool for load testing, performance measurement, and functional testing.

#### 4.7.8 Operational considerations

The time division multiplexor interface between CUESTA and the SUT makes it possible to emulate a large number of low-speed asynchronous terminals without the logistical problems associated with the usual requirement of one RTE/SUT cable per terminal. Only one cable per time division multiplexor is required, and it can be shared by as

many as 52 emulated terminals. Consequently, 100 to 300 terminals can be emulated with six RTE/SUT cables. As will be discussed in section 5, the implications of emulating asynchronous terminals in this manner are not clear.

#### 4.8 IBM's Data Base/Data Communications Driver System (DB/DC Driver)

The Data Base/Data Communication Driver System (DB/DC Driver) is one of the two RTE's developed by IBM. It is a tool for testing and driving one or more communication oriented or data base application programs.

##### 4.8.1 General description

One feature of this RTE is that it operates in one of two modes. In simplex mode, the RTE runs internal to the teleprocessing System Under Test (SUT). In duplex mode, the RTE resides in a CPU external to and independent of the SUT. Duplex is the mode of operation of concern in this report; however, it is interesting to note that both types are available in one implementation.

Another interesting feature of this RTE is the logical driver concept. A logical driver is a subset of the driver system resources. Each logical driver is controlled by a test supervisor; each test supervisor may interact directly with the driver system in real time application via a logical driver console. The driver system can be shared by up to 16 test supervisors, each in complete control of their own regions.

There is a library function. A user of the driver may use scripts from private or shared libraries. This type of capability eliminates the re-writing of scripts by groups of individuals interested in performing similar functions.

The script language provides for specification of emulated user think time. The think time for a specific transaction is determined by the think time distribution for the emulated device. There is not a command to regulate emulated user typing rate; rather, all characters are sent at the speed of the transmission line between emulated device and SUT.



#### 4.8.2 Hardware considerations

The configuration of DB/DC Driver is dependent upon the terminal requirements to be supported and the objectives of the installation. Real storage requirements are dependent upon the number and type of terminals to be emulated.

The minimum CPU on which DB/DC Driver can be implemented is an IBM 370/145. The communications front-end requirements are handled in one of two ways. Either one 3705 communications front-end is used for the driver system in addition to another communications front-end for the SUT, or one 3705 may be shared by the driver system and the SUT provided the Partition Emulator Program (PEP) is run in this 3705. The reason this sharing is possible is because the resources of a 3705 are greater than the demand placed on it by either the emulator or the SUT during a test.

Other hardware requirements include a printer output unit, a card input unit, a punched card output unit, and a magnetic tape unit. The operating systems which support DB/DC Driver are OS/VS2 Release 1.7 (SVS), OS/VS2 Release 2.3 (MVS) or OS/VS1.

The minimum storage requirement is 328K of virtual storage and 80K of real storage. This configuration will support the testing of one application program and ten emulated terminals. For each additional four emulated terminals, the virtual storage must be increased by 8K; if the added terminals are 3270 CRT terminals to be used in display mode, the increase must be 16K.

The devices which are emulated are the IBM 2741 Communications Terminal with Correspondence line code or PTTC/EBCD line code (EBCDIC), teletype-compatible units type 33 or 35, and the IBM 3270 Information Display System. In addition to the emulated devices, three real terminals may be used during application of the driver. The command console provides for real time operator communication with the RTE. The communications console is used to record system error messages and activity. The logical driver command console is used by a test supervisor to control script execution. A minimum of one logical driver console is required; a maximum of 16 is allowed. No more than 255 terminals can be simultaneously online with the driver system. This constraint includes consoles, other real terminals and emulated terminals.

DB/DC Driver interfaces through the communications front-end of the driver system to the communications front-end of the SUT such that no modifications to the SUT are required. The driver system includes a modified network control program which resides in its front-end. This

program communicates with the driver system as a network control program, and with the front-end of the program under test as would a terminal.

#### 4.8.3 Script preparation

The driver system uses two types of scripts which use the same script language, but differ in the type of entries in the script. The "master scenario" defines the logical driver and the network to be emulated through the use of parameters and relates the driver hardware to the terminal scripts. The "terminal script" contains the terminal activity to be emulated. Both script types reside as line-numbered files in the driver system. Terminal scripts are read, transactions built and transmitted a transaction or message at a time.

Scripts may be developed in several ways. There is a facility available in existing systems using either the Information Management System (IMS) or the Customer Information Control System (CICS) for the capture of real user/system dialogues. The actual transmissions from a real terminal to the SUT and responses from the SUT to the terminal are captured via the log file in IMS or via a pre-defined file in CICS. An automatic script generation program is then executed to produce scripts to be used in subsequent tests. This facility has been used to facilitate regression testing.

A second method of script development is the use of the entry and edit facilities of an on-line system such as TSO to produce the scripts. A utility is provided with the driver system to take these scripts files and load them into the driver data base for use in subsequent tests.

The third method is the use of the on-line entry and edit facilities of the driver system itself.

#### 4.8.4 Data acquisition

After an emulated device's message is transmitted to the application program under test, the transmission line is turned around to await the reply. When a reply is received, it is time-stamped and logged.

The log contains the transactions sent to the SUT, the responses from the SUT, and the time stamps.

#### 4.8.5 Data summarization

After a test, a utility routine converts the log file to a formatted data file. The driver system provides a utility program that will produce a large variety of reports from the log tape under control of user-specified parameters. Examples of the types of summarization reports possible are: average transaction processing time, poll delay, and response time. In addition, histograms may also be produced for given parameters for any number of terminals from one to all.

#### 4.8.6 Integrity confirmation

Because all transactions to and responses from the SUT are logged, all interactions between the SUT and DB/DC Driver can be printed for investigation following application.

#### 4.8.7 Current usage

DB/DC Driver is a marketable program product offered by IBM. IBM recommends the driver system for usage in five major testing areas: regression testing, functional testing, load testing, network emulation, and performance measurement testing.

#### 4.8.8 Operational considerations

If response checking is desired, there is an optional user defined table which can be specified. Each entry in the table contains up to 16 characters of a possible reply from the program under test. The entry also contains the code which defines the action to be taken. A user transaction is transmitted to the SUT; when a reply is received, it is compared to the entries in the "critical message table." If a match is found, the table-specified action is performed. If no match is found, a determination is made if an entire message has been received. If not, more information is read. Otherwise, the next emulated user transaction is sent.

#### 4.9 IBM's Teleprocessing Network Simulator (TPNS)

The Teleprocessing Network Simulator (TPNS) is one of two RTE's developed by IBM. TPNS was designed to provide controlled generation of message traffic into a telecommunications subsystem or application.



#### 4.9.1 General description

TPNS performs many functions: emulates terminals, emulates a communication network, generates message traffic on existing lines, drives application programs, captures and time stamps messages and responses, and reports results. Like IBM's DB/DC Driver and some other RTE's, TPNS runs in one of two modes. In simplex the RTE runs internal to the System Under Test (SUT). In duplex the RTE runs in a CPU external to the SUT.

TPNS can drive multiple applications through multiple communications front-ends via multiple networks concurrently. TPNS is hierarchical in design; the network is emulated as lines connected to multiple terminal controllers connected to individual devices. "Network" refers to the hardware and message traffic that TPNS emulates. "Line" refers to the teleprocessing link between SUT and emulated terminals. "Terminal" corresponds to the controller attached to the teleprocessing link. "Device" denotes a work station attached to a controller. In addition, control of message traffic rates may be specified to TPNS at any of these levels of the network: network, line, terminal, and device.

TPNS emulates think time for emulated user inputs in one of two ways. Think time may be specified as the interval of time between receiving a SUT response at the emulated terminal and the input of the next user message. Alternatively, think time may be specified such that emulated terminal messages are sent to the SUT at designated intervals irrespective of the SUT response; this specification was developed primarily for use of query-based systems in which queries are sent to the SUT without waiting for responses to previously issued queries. Since the emulated devices are treated in a polled environment, if the designated interval is selected such that it is shorter than the polling interval, the emulated terminal messages must wait the minimum amount of time imposed by the poll before the message is sent. There is not a script command to control typing rate.

#### 4.9.2 Hardware considerations

TPNS can be minimally configured to run on an IBM 370/145 with conditional swapping feature. TPNS executes under all currently available IBM 370 operating systems. The minimum virtual region size of 320K will drive a network of up to five lines and five terminals. One tape drive is required if data logging is desired. One 3705 must be dedicated to TPNS during tests; it requires a minimum of 80K storage.

The asynchronous terminals supported are the 1050, 2740 Model 1, 2741, AT&T 83B3, and Western Union 115A. The Binary Synchronous terminals supported are the 3270 Models 1 and 2. Additionally BSC 1, 2, and 3 are supported line disciplines. The BSC terminals supported include the 1130, 1800, 2770, 2780, and the 3780. Synchronous Data Link Control (SDLC) line discipline for Systems Network Architecture (SNA) terminals are supported. These include the 3600, 3650, 3660, 3614, 3767, 3770, 3790, and 3270.

TPNS interfaces through the 3705 communications front-end of the driver system to the communications control unit of the SUT. The TPNS 3705 performs part of the driver functions of TPNS. The communications front-end of the SUT may be either a 270x or a 370x.

#### 4.9.3 Script preparation

Emulated input message data is generated in several ways. It can be explicitly given in script files. However, there are several other interesting options. Text can be obtained internally from tables, a random number generator, a time/date/sequence source, terminal ID, a segment from a buffer of text received by an emulated device such as the 3270, and terminal action keywords. Text may also be obtained externally from a message file.

#### 4.9.4 Data acquisition

All messages are time-stamped when received or sent by TPNS and logged on magnetic tape.

#### 4.9.5 Data summarization

There are four data-related outputs: interval reports, a log tape, a log listing, and response analysis. Interval reports are based on data collected on a periodic basis. This information may be printed in real-time or spooled and printed later. Response analysis is done by listing response times and the number of transactions having that associated response time.

The statistics are calculated by communication line and include the average, high, low, and 90th percentile. These response times may be selected as internal host processing time or total user-observed response time at the emulated terminal.

#### 4.9.6 Integrity confirmation

Because all messages are time-stamped and logged, the procedure of studying the log of all interactions following a test can be used to verify the execution of the scripts by the SUT. In addition, real terminals may be connected to the lines on which TPNS emulated terminals are transmitting to observe and manually time responses at the real terminals.

#### 4.9.7 Current usage

TPNS is a marketable program product offered by IBM. It provides a repeatable environment for functional testing, regression testing, evaluation of network design, and approximation of application performance.

#### 4.9.8 Operational considerations

There is a preprocessor which provides syntax checking of scripts before execution. There is a set of operator commands which may be issued at the control console during execution of TPNS; included in this set are commands to modify emulation parameters and invoke terminal recovery facilities, and change message transmission rates during the test.

TPNS is started as a normal job under the operating system. During start-up the TPNS NCP is loaded. Following this procedure, the network initialization is performed. Based on the user specifications in the network configuration cards and message generation cards, the network is built.

TPNS is a sophisticated system which includes logic test to provide for content analysis of SUT messages. This facility enables logical responses from emulated terminals based on SUT responses.

#### 4.10 Tymshare's MEDLINE Simulated User System (MSUS)

MEDLINE is an international information system to support the delivery of health care services and the education of health care professionals and related personnel. MEDLINE was implemented by the National Library of Medicine (NLM) to run on the IBM 360/370 family. It is currently running on an IBM 370/158. The majority of MEDLINE users access this information system via the Tymshare Network (TYMNET). NLM contracted Tymshare to



implement a system to emulate user interactions with MEDLINE in such a way that the communications front-end could not distinguish such interactions from those with real users. This system was needed by NLM to insure that planned developmental changes to MEDLINE would result in improved, not degraded, system performance.

Originally, Tymshare implemented the MEDLINE Simulated User System (MSUS) under the Single Simulated User Program (SSUP). As the name implies, SSUP was only able to emulate one user. The Multiple Simulated User Program (MSUP) has replaced SSUP in MSUS. Because MSUP is an extension of the previous implementation, it alone will be discussed.

#### 4.10.1 General description

MSUS is unique in the context of the other RTE's because it represents an approach to remote terminal emulation in which the emulator is always available on-line to users of a computer network. There is no special hardware for the user to configure; rather a simple command, MSUP, is issued at the user's terminal and the RTE is available. Because Tymshare is also providing computer service to persons other than those using the RTE and because execution of MSUP does add an increased load to the host on which it runs as well as the network, the number of terminals which can be emulated is limited by the workload on the Tymshare host computers at the time it is initiated.

There is a script command to control the typing rate for each emulated device. There is also a command for think time. Additional script commands include a command for imposing pauses within the user messages and a command for error handling.

#### 4.10.2 Hardware considerations

MSUP is able to run on any one of the XDS 940 host computers on the Tymshare network. It communicates with the System Under Test (SUT) through the communications lines provided by the network between the 940 running MSUP and the SUT.

A maximum number of 20 emulated devices per XDS 940 may transmit messages at 110, 150, and 300 bits per second. The National Library of Medicine has used three 940's simultaneously to emulate 60 users.

#### 4.10.3 Script preparation

Scripts are created in the Tymshare LNED package. LNED is an easy-to-use line editor which permits the user to enter, delete, or modify lines simply by entering the number of the line and the desired contents. Once files are created, they may be changed simply by invoking LNED.

#### 4.10.4 Data acquisition

The data descriptive of the RTE/SUT interactions are recorded in the Interaction File. There are three distinct general classes of information records found in this file. The Characteristic Record occurs only once in every Interaction File, and it contains the general descriptive information regarding the session: for example, the number of users logged into MEDLINE at initiation, the date and time of the test, emulated terminal transmission rate, and emulated user typing rate. The User Interaction Record occurs for each line of the script file read. This record contains such information as the elapsed time required to transmit the interaction at the specified typing rate, the elapsed time between sending the last character of an emulated user message and receiving the first character of the SUT's response, the elapsed time to receive the response at the emulated transmission speed, and the text of the user's interaction. The System Interaction Record which follows each User Interaction Record contains operation numbers associated with each type of interaction.

Rather than performing the time-stamping at the 940 running MSUP, it is performed by the SUT. This is accomplished by MSUP sending requests to the SUT for the amount of resources utilized by an emulated terminal. This implies that only the SUT's performance is being tested, without regard to the communications network.

#### 4.10.5 Data summarization

Several reports and analyses are provided by the Multiple Simulated-User Reports (MSUR) program. Output from this program can either be directed to the user's terminal or to a file. There are three reports which can be generated according to the desires of the MSUR user.

MEDLINE User System Elapsed Time, MUSET, prints the elapsed time listing for any specified test. The user identifies the Interaction File name and the specific operation numbers to be sought. There are four types of elapsed times which are available: the time to transmit emulated user messages, the response time, the time to print

the SUT's response, and the total interaction time.

MEDLINE User Simulator Analysis and Distribution, MUSAD, has three options: elapsed time analysis, elapsed time distribution, or both. Again, one of four elapsed times is available for calculation: transmit, response, SUT print, and total interaction times. The MUSAD user specifies which operations are to be represented as columns in the print-out. The elapsed time analysis option displays the number of Interaction Records which fall into specified time intervals by column, where each column represents a specified operation. The Elapsed Time Distribution option displays the number of Interaction Records occurring in each interval as a percentage of all Interaction Records for that column. If the combined report is selected, then both the count and percentage reports are printed.

MEDLINE User Simulator Source Comparison, MUSSC, isolates any output differences between two tests. The user identifies the Interaction Files to be compared, and the specific operations. If differences are found, the complete interaction is printed, including user and system responses. The total number of differences is also calculated.

There are also several programs which were written by the staff of the National Library of Medicine. These reports concentrate on average response times and factor out the communications network components; that is, these reports strictly relate to the performance of the host computer system.

#### 4.10.6 Integrity confirmation

MUSSC, described above, provides a check on script execution. When identical scripts produce different results, the entire interactions are printed to aid in determining the reason for the differences.

#### 4.10.7 Current usage

MSUP is currently used by the National Library of Medicine for performance testing of its on-line services. In addition, a copy of MSUP is available for another Tymshare user for contract monitoring activities. If the user finds that service falls below a specified minimum over a specified period of time, the contract contains a clause for appropriate action by Tymshare.



#### 4.10.8 Operational considerations

MSUP is available as any application program on a Tymshare 940 computer. Because hundreds of users may be served by Tymshare simultaneously, it is necessary to provide MSUP with the capability to reject a proposed test. This reject capability is required to insure that the service quality of the Tymshare users not concerned with the test may continue at an undegraded level. MSUP users must specify the number of terminals to emulate; if the workload is such that the additional load would cause degraded service to other users, MSUP suggests a lower number to emulate.

There is an interesting script command which informs MSUP of the maximum interval of time to wait for the SUT's response to a message sent by an emulated user. If the maximum is exceeded during any test, the user of MSUP is informed, and the test is aborted for that one emulated device. Again due to the on-line environment of MSUP usage, this command is especially useful. While the network may not be heavily loaded at the beginning of a test, the network workload may increase during the test and with the extra burden of running MSUP, the test may be invalidated for one or more emulated devices.

#### 4.11 Univac's Communications Network Emulator (CNE)

Univac has developed the Communications Network Emulator (CNE) as a tool for use only within the company; its primary application is in support of marketing at the test center in Minneapolis.

##### 4.11.1 General description

CNE runs in any model Univac 1100 series computer. It may be operated as an RTE, or may be operated as an internal measurement driver. Operator think time is emulated; it is specified by a special script command. Transmission of messages is performed at the maximum line rate for asynchronous as well as synchronous transmission.

##### 4.11.2 Hardware considerations

CNE emulates Univac product line CRT terminals operating in asynchronous or synchronous multiplex mode, terminal work stations involving card and paper tape facility, asynchronous keyboard hard copy terminals, and terminal multiplexers. It is capable of emulating

synchronous and asynchronous communication processors with varying lines speeds. It is designed to run a maximum of 32 lines concurrently; multiple copies of CNE must be loaded in the driver system to run additional modules of 32 lines.

The line configuration may consist of any permissible configuration of the corresponding Univac hardware, including single and multi-drop networks and cascaded multiplexers. The network emulated is described by a number of optional card images which specify the terminal type and parameters.

CNE checks all messages output from the System Under Test (SUT) and waits for the appropriate prompt character, which differs according to whether the emulated terminal is synchronous or asynchronous. It also checks for a repeat request message, which is again different for synchronous and asynchronous terminals, and takes appropriate action by retransmitting the last message.

#### 4.11.3 Script preparation

Scripts are constructed using one of the standard input modes available on the Univac 1100, such as editor or punched card input. These scripts contain exactly the same input as would be produced by terminal users, augmented with a few commands in the standard command language format, which are specific to CNE usage.

The special commands include the following: an indication that the following terminal input is emulated paper tape or card input; a number of commands to set and reset timers and to recognize certain characters as solicitation and termination characters; an ability to set the think time; and an ability to cause a specific card further down in the run stream to be transmitted at a specified time relative to the present time. There is also a repeat count command which allows the restarting of scripts from lines 0 to 8.

#### 4.11.4 Data acquisition

All input and output buffers and error messages are recorded on mass storage for subsequent analysis by programs which are provided as part of this package.

CNE timestamps the first character transmitted from buffered terminals which would be the time at which the user presses the transmit key. For asynchronous devices, CNE timestamps the last character, the carriage return ending input. System output is timestamped with the calculated

time of the first character transmitted for synchronous communication which occurs in fixed block lengths. In asynchronous communication, system output is timestamped with the first character transmitted. There is no capacity to provide timing for a live terminal. It is difficult although not impossible to display the traffic between the SUT and an emulated terminal.

#### 4.11.5 Data summarization

There are two related data analysis programs. One allows the data analysis user to select an individual terminal or a set of terminals, and to print out a complete history of all transactions occurring between the SUT and the emulated terminal, or a history of error messages, or a history of all text messages. The other data analysis program provides a listing of all transactions for every active terminal followed by a device summary which gives statistical information on that device. After all terminals have been listed, a network summary is given containing the same statistics. Statistics include averages, medians, and standard deviations, for individual devices and the network. Optional controls permit the elimination of any or all of these statistics as well as the text data printouts. In addition, this program allows users to provide a string of 8 characters to uniquely identify a given user input for which statistics will be calculated. Another option allows the user to specify the start and stop time over which statistics are to be collected.

#### 4.11.6 Integrity confirmation

The procedures which can be used for integrity confirmation are available as post processing activities. One is the printed record of all SUT/RTE transactions. This record can be studied to determine if the interactions proceeded as expected. The history of error messages also provides a means to insure that the SUT and RTE functioned properly during a test.

#### 4.11.7 Current usage

At the present time CNE is available only for use internal to Univac. It is used in their responding to benchmarking requirements.



#### 4.11.8 Operational considerations

For small network configurations, the description of the lines and terminals may be prepared by the user directly. For assistance in configuring large networks, there is a COBOL program which will produce a file of the necessary card images which can then be used as input to CNE.

#### 4.12 Univac's Communications Simulator (CS1100)

Univac has an RTE called Communications Simulator, CS1100, which is available to all Univac 1100 installations at no charge.

##### 4.12.1 General description

CS1100 provides the user with the capability of emulating a complete communications environment without requiring terminal operators or terminals. CS1100 is provided in three major modules: traffic control language (TCL), communications line simulator (CLS), and remote terminal simulator (RTS). This division into modules conveniently isolates the functions of emulating terminal operators, emulating communications, and emulating the terminals. The aggregate of these modules constitutes the RTE.

TCL is a language developed to control traffic between an emulated remote terminal and the System Under Test (SUT). Each emulated terminal is controlled by a TCL program that describes the actions of the operator of the emulated terminal. There is a command to emulate think time. There is not a command to control the emulated user typing rate; characters are transmitted at the speed of the communications line. TCL programs may read scripts from program or data files, send messages from an emulated terminal to the SUT, and receive messages from the SUT. The TCL language, which bears a strong resemblance to SNOBOL, contains a set of powerful and flexible commands used to examine and change images received from the SUT or read from the files and thus control the action of the emulated terminal operator.

CS1100 can operate as an internal driver, without use of communications hardware. When so operated, CLS is employed to emulate this hardware. CLS is the aggregate of the 1100 series executive changes required to achieve emulated communications line data transfer. Since it is used only when CS1100 is operated as an internal driver, it

will not be discussed further here.

RTS is a real time user program which emulates the hardware remote terminals. For each type of remote terminal to be emulated, RTS incorporates a hardware device emulator. RTS then provides the interface between each hardware device emulator and the TCL dispatcher which controls the execution of TCL programs.

The actions of RTS are directed by control statements which are provided in a file, from an interactive terminal, or from the on-site operator console. The available commands may be divided into three broad categories: 1100 Series operating system commands, commands affecting the operation of the RTS emulation, and specification commands describing the terminal network to be emulated. The commands affecting the operation include a command to display the count of the number of active terminals, a command to display the memory usage of CS1100, a command to set the delay rate between the reading of control statements from a file, a command to insert a wait just before the next control statement is read, as well as a command to terminate one or more emulated terminals.

TCL allows RTS to be a programmable RTE. The code generated by the TCL compiler is loaded by RTS into itself. Subsequently, control is dispatched between the hardware device emulator and the TCL program by the RTS dispatcher.

TCL is a language for ASCII text manipulations, plus a few special statements for interfacing with RTS. Several of the features of the SNOBOL language are present, along with arithmetic and logical operation on integers. TCL is a special purpose high level language designed to control and analyze traffic between emulated remote terminals and SUT's. TCL programs may operate in conjunction with one or more transaction control files which contain images that may be sent to the system by the emulated user or used as parameters to the TCL program. Thus, the TCL program may either send images from the transaction control file or generate and send its own control files images. The absolute program produced by the TCL system program is reentrant so that any number of terminals may be emulated by the addition of just the data required for that terminal.

#### 4.12.2 Hardware considerations

The supported terminals include asynchronous ASCII teletypewriter terminals; full-duplex remote synchronous batch terminals (reader, printer, punch); synchronous CRT terminals, configured in single station multidrop, multiplex, cascaded multiplex, or multidrop multiplex

networks; and keyboard reader printer punch synchronous workstations.

CS1100 is designed to support additional terminals as they are added to the Univac product line, it is therefore easy to install drivers for additional terminals. Users with other terminals may add their own drivers so that these terminals may also be emulated.

#### 4.12.3 Script preparation

Scripts to control the operation of CS1100 may be stored as symbolic entities (called "elements" in CS1100 terminology) in the RTE computer's file system. These scripts may be input from cards or from a terminal and may be modified using any of the available editor-like subsystems.

In order to simplify the generation of RTS control language statements necessary to configure large networks, a skeleton is provided to generate the device specification control statements.

#### 4.12.4 Data acquisition

RTS maintains an audit trail of all communications activity in a log file. The log entries include: the complete input buffer, the complete input message, the complete output message, the complete output buffer, time stamps, terminal initialization and termination, RTS initialization, date change, and TCL print statements.

The time stamp associated with each message is the time that the message has completed transmission; that is to say, the time of the last character. In the case of synchronous lines, input to the SUT is time stamped upon the acknowledge from the SUT.

#### 4.12.5 Data summarization

A utility processor is made available for reading and printing the log file. The available options include: print a batch format listing; print log entries and headers; print TCL program statements; print response times; print the read and print log entries only; and print response time graph. Output may be selected for a specific terminal or for all terminals. When a specific terminal is selected, printout includes the time stamp for input (emulated user messages sent to the SUT) and output (SUT messages sent to emulated users), and the time



difference between input and output. Due to the method of timestamping, the response time is the time from the last character of an emulated user message to the last character of the corresponding SUT message.

#### 4.12.6 Integrity confirmation

Scanning the listing of all characters exchanged between emulated devices and SUT verifies that the scripts have executed correctly. The option of selecting the summary of a single device provides the added check on timing specifications (e.g. are the delay or think times of the correct duration?).

#### 4.12.7 Current usage

CS1100 is used by Univac in the regression testing of the 1100 Series Executive communications handler, the communications control routines (CCR's), and the 1100 Series TIP/CMS transaction system.

CS1100 is available to all Univac 1100 series installations at no charge.

#### 4.12.8 Operational considerations

CS1100 has been designed as a modular system and documentation is available on the three modules: TCL, CLS, and RTS. The message traffic controller, TCL, may run with or without the emulator of the terminal devices, RTS. Use of TCL without RTS is more suitable to in-house testing than to the procurement environment. In procurements the emulation of the hardware is a significant factor.

## 5. Problem areas

As mentioned in Section 2, the issues of repeatability and flexibility impact the use of RTE's. The tradeoffs between these two issues are weighed by the purpose of the RTE application. For in-house testing of a SUT, it appears that the need for flexibility is greater than in a procurement environment. That is, if the SUT malfunctions from overload during a procurement Live Test Demonstration, the vendor will probably decide to reconfigure a SUT for a new test. In in-house testing, the RTE should be able to field operator messages indicating that the system is under stress and log these messages for later analysis. System operators can use such inputs for performance reports and determining system bottlenecks.

Emulating a large number of asynchronous terminals is a problem for some of the RTE's studied. The definition of "large" varies, but is apparently between 50 and 100. The difficulty appears to be more fiscal than technical. While there is no difficulty in configuring a system for installation at the customer's site, there is a problem with the test system at the vendor's benchmarking facility. There is a cost, on the order of \$750 to \$2,000, in configuring each line between the RTE and SUT. Non-multiplexed asynchronous terminals each require one line, thereby significantly impacting the cost of setting up an emulation.

For in-house testing of a system, multiplexing asynchronous devices is not a prohibitive feature. That is, the system is tested prior to system changes to determine performance; then, following system changes it is tested using the same configuration to determine if any degradation has occurred. However, in the procurement environment the impact of configuring asynchronous devices is not clear. For example, after delivery of the computer system when the asynchronous devices are configured on the basis of one terminal per communications line, the system may not deliver the same level of service. Of course as long as the service is within the RFP requirements, it is acceptable. There is currently not agreement as to how to alleviate this problem.

Allowing a vendor to substitute equipment other than that specified is not an uncommon practice. Questions are raised as to the representativeness of the test in which a substitution is made, whether the workload on the SUT is increased or decreased by the substitution, and how the SUT should be configured relative to the equipment required for the live test demonstration. The alternatives include the use of keyboard interactive terminals operating in synchronous multidrop or multiplex mode, asynchronous terminal traffic already processed through a multiplexer,

and connection of RTE and SUT by a data path which does not follow any communications line discipline.

Another problem is proper emulation of the data handling capacity of terminals. For example, at an RJE terminal the limiting factor on throughput could be the speed of the line printer. Therefore, scenarios should not specify unrealistic performance of the emulated device.



## 6. Summary

The trend in computer usage has been from on-site batch processing systems to systems which encompass on-site batch, remote batch, and interactive teleprocessing. Batch programs when timed by observers using stop-watches constituted adequate benchmarking methodology for on-site batch systems. Benchmarking of teleprocessing computer systems requires different techniques to evaluate the system quality of service. Remote terminal emulation is an approach to such evaluation.

This report has discussed key factors in remote terminal emulation that were identified through personal discussions with vendors and implementors of RTE's, site visits, and interactions with users of RTE's. Twelve RTE's were identified for which detailed information was available. These RTE's were described to indicate the current capacity and capability of these devices, and implementation considerations.

The majority of RTE's were originally developed for in-house testing by vendors of computer systems and services. For the most part, vendors recognized the need for a tool to be used for such activities as the stress testing, software debugging, and tuning of teleprocessing systems. As the RTE's matured and their use became recognized, in some companies they became marketing tools, and, in some cases, marketed software products. It is interesting to note that while vendor-developed RTE's were designed independently, they are functionally very similar.

-----

The authors acknowledge the excellent cooperation of the implementors of the RTE's described in this survey, both in providing information and validating the technical content of this report. Specifically, the assistance of the following persons is acknowledged: Air Force (Ray Carle), Burroughs (Bill Bird, Chris Clark, Bill Johnson), Control Data Corporation (Will Kane, Bob Korsch, Juris Skujins), Digital Equipment Corporation (Dave Kearsis, Ed McHugh, Rollins Turner), Hewlett Packard (David Crockett, Ed Giamo, Mike Jones, Len Shar, Chris Wilson), Honeywell (Couley Arthur, Israel Epstein, Jack Keim, Clare Miller, Dick Quinn), IBM (Jerry Boyle, Tom Eccles, Bob Follett, Lydia J. Friedman, Ray Youstra), MITRE Corporation (Dave James, Dave Lambert), National Library of Medicine (Barbara Sternick), Tymshare (Guy Blood, Ron Stevens), and Univac (S. Bolin, Larry Hakeman, Jack High, Larry Sheets).

## References

Abrams and Cotton[1975]

Abrams, Marshall D. and Ira W. Cotton, "The Service Concept Applied to Computer Networks," NBS Technical Note 880, August 1975, 38p.

Crothers[1976]

Crothers, C. G., "Testing TSO with a Remote-Terminal Emulator: Some Preliminary Results," the MITRE Corp., MTP-168, January 1976, 25p.

Drummond[1973]

Drummond, Mansford E., Jr., Evaluation and Measurement Techniques for Digital Computer Systems, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973, 338p.

GSA[1975]

General Services Administration, Automated Data and Telecommunications Service, Procurement Memorandum Number 3, November 5, 1975, 2p.

Hawkes[1975]

Hawkes, John, "TEPE Time-Sharing Event Performance Evaluator User Guide," a Hewlett-Packard internal document, June 1975.

IBM[1975]

IBM, "DB/DC Driver System General Information Manual," September 1975, 35p.

IBM[1976]

IBM, "Teleprocessing Network Simulator (TPNS) Advanced Language Guide," January 1976, 150p.

IBM[1976]

IBM, Teleprocessing Network Simulator (TPNS) Base Language Guide," January 1976, 120p.

IBM[1976]

IBM, "Teleprocessing Network Simulator (TPNS) Program Reference Manual," January 1976, 162p.

Kosko and Turner[1975]

Kosko, David and Rollins Turner, "Data Transaction Processing Experiment on the PDP-11/70," Report Number 75DK378440, November 5, 1975, 8pp.

Lockett and Bell[1975]

Lockett, Jo Ann and Thomas E. Bell, "The Rand Monitor/Stimulus Generator: A Simple Approach to Response Time Analyses in On-Line Computer Systems," Rand Report R-1649-PR, May 1975, 26p.

MITRE Corp.[1973-1975]

MITRE Corp. "Remote Terminal Emulator (Design Verification Model)," Bedford, Mass., Technical Report MTR-2677, Volumes as cited below:

Vol. 1, " - Introduction and Summary," February 1975, 76p. (By D. L. James and D. W. Lambert).

Vol. 2, " - Data Structures, Scenario Instructions and Commands," November 1973, 79p. (By D. L. James).

Vol. 3, " - Macro Preprocessor," November 1973, 47p. (By M. L. Dolley).

Vol. 4, " - Scenario Assembler," August 1973, 83p. (By Tobyanne Paster).

Vol. 5, " - Scenario Interpreter," April 1974, 227p. (By D. L. James).

Vol. 6, " - Real-Time Executive," July 1975, 158p. (By M. J. Sullivan).

Vol. 7, " - Data Reduction Program," April 1974, 129p. (By M. C. Sarajian).

Vol. 8, " - Description of Hardware," November 1974, 76 p. (By E. C. DeMone).

Vol. 9, " - Support Software and Hardware Diagnostics," December 1973, 48p. (By M. C. Sarajian).



Peterson[1974]

Peterson, K. A., "Remote Terminal Emulator Test Scenarios," Bedford, Mass., Technical Report MTR-2776, 141p. August 1974.

Sperry Rand[1976]

Sperry Rand Corp., "CS1100 Communications Simulator Reference Manual," January 30, 1976.

Sullivan[1975]

Sullivan, M. J., "Validation of Teleprocessing Networks in the Competitive Procurement Environment," Bedford, Mass., Technical Report MTR-3012, 48p., April 1975.

Suyemoto[1975]

Suyemoto, T., "Data Reduction Considerations for Remote-Terminal Emulator," Bedford, Mass., Working Paper WP-20155, 22p., February 1975.

Turner and Levy[1976]

Turner, Rollins and Hank Levy, "Performance Evaluation of IAS on the PDP-11/70", Proceedings of the International Symposium on Computer Performance Modeling, Measurement and Evaluation, March 29-31, 1976, Harvard University, Cambridge, Massachusetts, pp. 71-74.

Yoshimura[1975]

Yoshimura, Roy I., "The Rand Monitor/Stimulus Generator: Hardware Implementation," Rand Report R-1716-PR, November 1975, 25p.

# APPENDIX A

## Communications Disciplines Supported

RTE	Asynchronous	Synchronous
Air Force/MITRE DVM	*	*
Burroughs System/DCEM	*	*
Control Data Corp. BARTER	*	*
Digital Equipment Corp.	*	
Hewlett Packard TEPE	*	
Honeywell CUESTA	*	*
Honeywell DATUS	*	
IBM DB/DC Driver	*	*
IBM TPNS	*	*
Tymshare MSUS	*	
Univac CNE	*	*
Univac CS1100	*	*

# APPENDIX B

## Maximum Number of Devices Emulated

RTE	Asynchronous	Synchronous
Air Force/MITRE DVM	64 @ 110 to 134.5 bps and multiples of 150 bps for 150 to 19.2K	64 @ 2.4K to 153.6K bps
Burroughs System/DCEM	No Known Upper Limit	
Control Data Corp. BARTER	64 @ 110 or 150 bps	8 @ 9.6Kbps or 16 @ 4.8K
Digital Equipment Corp.	32 @ 110 bps to 2.4K bps	
Hewlett Packard TEPE	32 @ 110 bps to 2.4K bps	
Honeywell CUESTA	96 communication lines at 110 to 9.6K bps up to maximum of 32 devices/line plus 300 time division multiplexed users @110 bps or 100 @ 300 bps	
Honeywell DATUS	17 @ 300 or 37 @ 150 bps or 52 @ 110 bps	
IBM DB/DC Driver	255 devices	
IBM TPNS	No Known Upper Limit	
Tymshare MSUS (per 940 computer)	20 @ 110 300 bps	
Univac CNE	32 communication lines	
Univac CS1100	No Known Upper Limit	



## APPENDIX C

## Emulated User Features

RTE	Think Time	Typing Rate
Air Force/MITRE DVM	*	
Burroughs System/DCEM	*	
Control Data Corp. BARTER	*	*
Digital Equipment Corp.	*	
Hewlett Packard TEPE	*	*
Honeywell CUESTA	*	
Honeywell DATUS	*	
IBM DB/DC Driver	*	
IBM TPNS	*	
Tymshare MSUS	*	*
Univac CNE	*	
Univac CS1100	*	

## APPENDIX D

## Measures of SUT Performance

RTE	Response Time	Other
Air Force/MITRE DVM	*	*
Burroughs System/DCFM	*	
Control Data Corp. BARTER	*	*
Digital Equipment Corp.	*	*
Hewlett Packard TEPE	*	*
Honeywell CUESTA	*	*
Honeywell DATUS	*	*
IBM DB/DC Driver	*	
IBM TPNS	*	*
Tymshare MSUS	*	*
Univac CNE	*	*
Univac CS1100	*	

## APPENDIX E

## Data Summarization

RTE	Basis of Summary Statistics	
	Per Terminal	Cumulative
Air Force/MITRE DVM	*	*
Burroughs System/DCEM	*	*
Control Data Corp. BARTER	*	*
Digital Equipment Corp.	*	*
Hewlett Packard TEPE	*	*
Honeywell CUESTA	*	*
Honeywell DATUS	*	*
IBM DB/DC Driver	*	*
IBM TPNS	*	*
Tymshare MSUS	*	*
Univac CNE	*	*
Univac CS1100	*	*

"Cumulative" includes data summarization of all emulated devices or on a selected portion of all devices.



# APPENDIX F

## RTE Verification Procedures

RTE	Output of all Interactions	Real-time Comm. line selection for monitoring	RTE Self-Monitoring	Real terminal Attached with data collection	Real terminal Attached without data collection	Log tape
Air Force/MITRE DVM	*	*	*		*	*
Burroughs System/DCM	*	*		*		*
Control Data Corp. BARTER	*	*		*		*
Digital Equipment Corp.	*		*			*
Hewlett Packard TEPE	*	*	*			*
Honeywell CUESTA	*	*				*
Honeywell DATUS	*	*				*
IBM DB/DC Driver	*				*	*
IBM TPNS	*				*	*
Tymshare MSUS	*	*		*		*
Univac CNE	*					*
Univac CS1100	*					*



U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET	1. PUBLICATION OR REPORT NO.  NBS SP 500-4	2. Gov't Accession No.	3. Recipient's Accession No.
4. TITLE AND SUBTITLE  COMPUTER SCIENCE & TECHNOLOGY:  Survey of Remote Terminal Emulators		5. Publication Date  April 1977	
		6. Performing Organization Code	
7. AUTHOR(S)  Shirley Ward Watkins and Marshall D. Abrams		8. Performing Organ. Report No.	
9. PERFORMING ORGANIZATION NAME AND ADDRESS  NATIONAL BUREAU OF STANDARDS DEPARTMENT OF COMMERCE WASHINGTON, D.C. 20234		10. Project/Task/Work Unit No.  6502125	
		11. Contract/Grant No.	
12. Sponsoring Organization Name and Complete Address (Street, City, State, ZIP)  Same as item 9.		13. Type of Report & Period Covered	
		14. Sponsoring Agency Code	
15. SUPPLEMENTARY NOTES  Library of Congress Catalog Card Number: 77-608013			
16. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)  This report describes twelve Remote Terminal Emulators (RTE's). The key terminology associated with remote terminal emulation is defined and possible application areas are discussed. Technical implementation details and operational considerations are addressed for each RTE. Summary tables are provided to indicate current RTE capabilities and capacities, as claimed by the RTE developers.			
17. KEY WORDS (six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)  Evaluation; interactive; measurement; performance evaluation; performance measurement; remote terminal emulation; remote terminal emulators; teleprocessing.			
18. AVAILABILITY  <input checked="" type="checkbox"/> Unlimited  <input type="checkbox"/> For Official Distribution. Do Not Release to NTIS  <input checked="" type="checkbox"/> Order From Sup. of Doc., U.S. Government Printing Office Washington, D.C. 20402, SD Cat. No. C13.10:500-4  <input type="checkbox"/> Order From National Technical Information Service (NTIS) Springfield, Virginia 22151		19. SECURITY CLASS (THIS REPORT)  UNCLASSIFIED	21. NO. OF PAGES  80
		20. SECURITY CLASS (THIS PAGE)  UNCLASSIFIED	22. Price \$1.45





**ANNOUNCEMENT OF NEW PUBLICATIONS ON  
COMPUTER SCIENCE & TECHNOLOGY**

**Superintendent of Documents,  
Government Printing Office,  
Washington, D. C. 20402**

Dear Sir:

Please add my name to the announcement list of new publications to be issued in the series: National Bureau of Standards Special Publication 500-.

Name \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip Code \_\_\_\_\_

(Notification key N-503)





## PERIODICALS

**JOURNAL OF RESEARCH** reports National Bureau of Standards research and development in physics, mathematics, and chemistry. It is published in two sections, available separately:

• **Physics and Chemistry (Section A)**

Papers of interest primarily to scientists working in these fields. This section covers a broad range of physical and chemical research, with major emphasis on standards of physical measurement, fundamental constants, and properties of matter. Issued six times a year. Annual subscription: Domestic, \$17.00; Foreign, \$21.25.

• **Mathematical Sciences (Section B)**

Studies and compilations designed mainly for the mathematician and theoretical physicist. Topics in mathematical statistics, theory of experiment design, numerical analysis, theoretical physics and chemistry, logical design and programming of computers and computer systems. Short numerical tables. Issued quarterly. Annual subscription: Domestic, \$9.00; Foreign, \$11.25.

**DIMENSIONS/NBS** (formerly *Technical News Bulletin*)—This monthly magazine is published to inform scientists, engineers, businessmen, industry, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on the work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing.

Annual subscription: Domestic, \$12.50; Foreign, \$15.65.

## NONPERIODICALS

**Monographs**—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

**Handbooks**—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

**Special Publications**—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

**Applied Mathematics Series**—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

**National Standard Reference Data Series**—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a world-wide program coordinated by NBS. Program under authority of National Standard Data Act (Public Law 90-396).

## BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

**Cryogenic Data Center Current Awareness Service.** A literature survey issued biweekly. Annual subscription: Domestic, \$20.00; Foreign, \$25.00.

**Liquefied Natural Gas.** A literature survey issued quarterly. Annual subscription: \$20.00.

**NOTE:** At present the principal publication outlet for these data is the *Journal of Physical and Chemical Reference Data* (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St. N.W., Wash. D. C. 20056.

**Building Science Series**—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

**Technical Notes**—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

**Voluntary Product Standards**—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The purpose of the standards is to establish nationally recognized requirements for products, and to provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

**Consumer Information Series**—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

*Order above NBS publications from: Superintendent of Documents, Government Printing Office, Washington, D.C. 20402.*

*Order following NBS publications—NBSIR's and FIPS from the National Technical Information Services, Springfield, Va. 22161.*

**Federal Information Processing Standards Publications (FIPS PUBS)**—Publications in this series collectively constitute the Federal Information Processing Standards Register. Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

**NBS Interagency Reports (NBSIR)**—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services (Springfield, Va. 22161) in paper copy or microfiche form.

**Superconducting Devices and Materials.** A literature survey issued quarterly. Annual subscription: \$20.00. Send subscription orders and remittances for the preceding bibliographic services to National Bureau of Standards, Cryogenic Data Center (275.02) Boulder, Colorado 80302.

**U.S. DEPARTMENT OF COMMERCE**  
**National Bureau of Standards**  
Washington, D.C. 20234

OFFICIAL BUSINESS

Penalty for Private Use, \$300

POSTAGE AND FEES PAID  
U.S. DEPARTMENT OF COMMERCE  
COM-215



SPECIAL FOURTH-CLASS RATE  
BOOK

---