Measurement of Interactive Response Time

The following article is reprinted from "Federal Information Processing Standards Publication 57," dated 1978 August 1. Copies of the complete document are available from the National Technical Information Service, US Dept. of Commerce, Springfield, Virginia 22161 for a nominal charge. The full document contains an Executive Overview (1 page), a "Summary Guidance" section (3 pages), a Glossary (1 1/2 pages), and a Bibliography (3 pages) in addition to the material reprinted here.

Federal Information Processing Standards (FIPS) are developed by the National Bureau of Standards. In this case the FIPS is a guideline rather than an actual standard. Readers who are interested in obtaining further information about this guideline, or who wish their names added to a mailing list for distribution of future documents on the subject of "Measurement of Interactive Computer Service Response Time and Turnaround Time" are invited to correspond with:

Marshall Abrams
National Bureau of Standards
Technology B226
Washington, DC 20234

GUIDELINES FOR THE MEASUREMENT OF INTERACTIVE COMPUTER SERVICE RESPONSE TIME AND TURNAROUND TIME

1. Introduction

There is a need to measure the interactive computer service delivered to users through computer networks. This need arises in the selection, evaluation, operation, tuning and testing of interactive computer-based services. Definition of service quality is a complex question which is only partially answered today. Development and application of methodologies for service measurement is also a complex problem. This guideline addresses the parameters about which there is the most current knowledge and agreement: response time and turnaround time. It provides introductory information on and guidance for their measurement in terms of: conditions of measurement, definitions of these functional performance measures, and methodologies applicable to measurement and test.

1.1 Audience

These guidelines are designed for use by Federal officials and other employees who have responsibility for the specification, measurement, evaluation or selection of an interactive computer service.

1.2 Applicability

These guidelines address the type of computer utilization characterized by an interchange of input and output between a computer and a person utilizing a keyboard terminal. Among the names that are frequently applied to this type of usage are *interactive, conversational, demand,* and *transaction-oriented.* In reading this document, *interactive* should be read as implying any of the preceding terms.

Since one aspect of a computer system's performance may be evaluated in terms of the service provided to its user community, the functional performance measures presented in this document can be used to evaluate an in-house system in the same way as an outside service.

Interactive computer service is usually delivered to an end user through a dedicated or shared communications network which connects the terminal to the computer. The internal operation of the communications network, like the internal operation of the computer, is beyond the scope of this document. Rather, this document addresses the service delivered through the network to the terminal, (frequently) measured at the point where the terminal connects to the network.

1.3 Context

There is a large number of related topics which bear on the measurement, evaluation, selection, and operation of interactive computer services. A partial list of these related topics includes workload characterization, feasibility studies, benchmarking, remote terminal emulation, economic analysis, acceptance testing, tuning, and stress testing. These guidelines are intended to be read and utilized in the context of these other topics. Of necessity, the size and scope of this publication is limited; the reader is cautioned, however, that considerations beyond those discussed may bear on his actions. Remote batch service, for instance, will be addressed in a separate guideline.

1.4 Other relevant Federal documents

These guidelines are issued as a FIPS Publication in meeting the NBS responsibility to provide standards and guidelines for Federal use in the areas of computer performance measurement and teleprocessing.

The following list identifies some other relevant Federal documents available at the time of publication which should also be employed by Federal agencies in planning, selecting, and using this type of computer service.

Special Notice Concerning the Teleprocessing Services Program, General Services Administration, Automated Data and Telecommunications Services, April 1977.

Lessons learned about: Acquiring Financial Management and other Information Systems, Comptroller General of the United States, General Accounting Office, August 1976 (GPO stock number 020-000-00138-1).

Teleprocessing Services Program, Solicitation Number GSCCDPR-H-00011-N-5-28-76 (as amended), General Services Administration, April 14, 1976.

"Management, acquisition, and utilization of automatic data processing (ADP)." *Code of Federal Regulations*, Title 34, Chapter 282 (34 CFR 282). Previously published as *Federal Management Circular FMC 74-5*.

"Federal Property Management Regulations, Government-wide automated data management services." 41 CFR 101-32.

Guidelines for benchmarking ADP systems in the competitive procurement environment. Federal Information Processing Standards Publication (FIPS PUB) 42-1.

Policies for acquiring commercial products and services for Government use, OMB Circular A-76.

Major Systems Acquisitions, OMB Circular A-109.

[Further background information about the topics covered in this section (1.4) may be found in bibliographic citations 19 and 43.]

1.5 How to read and use this guideline

The following major sections of this document discuss two measures of interactive service performance, the conditions of measurement, and the methodologies for test and measurement. An overview of each section is followed by specific recommendations and discussion of salient points. Throughout the document, specific summary guidance is set in boldface italic type ***(this is boldface italic)***. This document is organized into three major sections—user service measures, conditions of measurement, and methods of measurement and analysis—each of which contains subsections which address specific topics in increasing depth. The reader is advised to read the document in its entirety before attempting to apply the guidance for a particular purpose.

2. User service measures

In order to promote competition in the supply of computer services, requirements should be stated in terms of system-independent functional specifications, i.e., "system performance" rather than design or equipment performance specifications. These specifications should include the objectives of the service and the underlying data processing requirements, as opposed to equipment performance specifications, which describe internal measures such as cycle time and instructions per second. Interactive turnaround time and response time are two functional specifications which are quantifiable measures of interactive computer service.

Other measures of interactive computer service which are not readily quantifiable have not been addressed in these guidelines. Two such measures are the "friendliness" or "human engineering" of the service, and the rate at which novices learn to use the service.

Functional performance measures should be expressed in terms meaningful to the use of the computer service in accomplishment of agency objectives. The complex interactions among the hardware and the software of the service computer(s) and the communications equipment can make specification of hardware characteristics essentially meaningless.

There are other related service measures which are not within the scope of these guidelines. For example, the data communications network may be evaluated independently of the computers and other data terminal equipment attached to it.

[Further background information about the topics covered in this section (2) may be found in bibliographic citations 17, 25, 30, 38, 39, 51, and 53.]

2.1 Response time

From an individual user's viewpoint, response time is one of the primary measures of the quality of interactive computer service. Measuring response time requires specification of: the precise definition employed, which responses are to be measured, the circumstances under which measurements are to be made, and the format in which the measurement results are to be presented.

A definition of response time applicable to most situations is the elapsed time from the last user keystroke (which terminates a service request) until the first meaningful system character is displayed at the user's terminal. This definition assumes no type-ahead. It is also necessary to specify the workload component which is being responded to. Different response times may be acceptable for different interactive tasks.

An illustration of the preferred definition of response time is given in Figure 1. Note that at the beginning of the computer output there are several non-printing characters which, by definition, are not part of the response. This sequence is provided for illustrative purposes only; it will certainly vary among different manufacturers' computers and even among identical hardware running under the control of different operating systems.

In specifying acceptable response, it is recommended that one or more classes of interactive tasks be specified to the degree required by the complexity of the workload. Each task may have a different response time requirement. Examples: "In the use of the text editor for insertions, deletions, or changes involving *n* characters or less, the response time shall be less than *w* seconds 50% of the time"; "In the compilation of PL/I programs of less than *m* lines, the response time shall be *x* seconds or less 90% of the time"; "Retrieval requests to a bibliographic information retrieval service in which *p* criteria (keys) shall all be satisfied in order for an item to be displayed shall be serviced with a response time of *y* seconds or less 95% of the time."

Other response time factors may also be considered. A uniform response time has come to be recognized as very important, in contrast to a response time that varies widely from transaction to transaction. Users seem to prefer a known uniform delay to an unknown variable delay, even if the mean of the former exceeds the latter. In some applications the response of the system on particular transactions can be artificially delayed when it falls below the desired value, to reduce the variability of response.

The definition of response time given above assumes a sequence of operations (transactions)

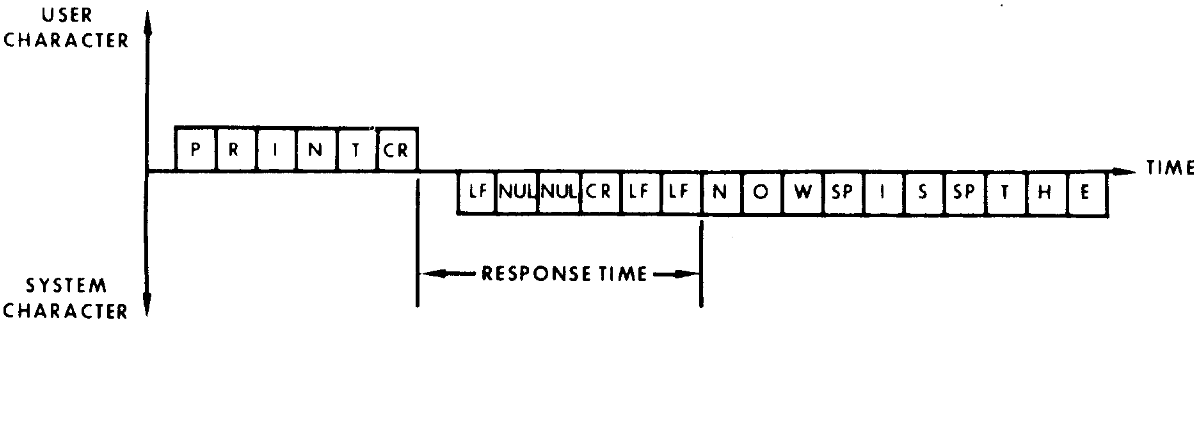

Figure 1. *Response Time*

composed of a user input followed by computer output which responds to that input. This sequence, illustrated in Figure 2A, assumes that the user waits until output is complete before entering the next input. This sequence is forced by some operating systems which prevent or ignore any attempted input until output is complete. An alternate sequence, shown in Figure 2B, allows the user to queue inputs. This mode of accepting multiple inputs to be processed in turn is most commonly known as "type-ahead."

The above definition of response time is not applicable when there is type-ahead. Type-ahead also introduces considerable difficulty in data analysis because of the overlap of input and output which may require a redefinition of elapsed time intervals. It is, therefore, recommended that type-ahead be excluded from any controlled testing. When measuring uncontrolled usage which includes type-ahead, careful thought should be given to the definition of response time and to implementation of data analysis programs. Since the definition and data analysis of response time are changed when type-ahead is employed, the use of turnaround time as the service measure is recommended when type-ahead is present.

Another complication arises when there is no local connection between the terminal keyboard and printer (or display) so that a character is visible to the user only when transmitted back from the network to the terminal (a process known as echoing). Echoed characters must be recognized in the data analysis as not constituting part of the response. Such recognition is extremely difficult, if not impossible, if the echoing is not exactly one-to-one. It is therefore recommended that, when echoes can not be automatically recognized by data analysis software, a non-echo mode of operation should be employed.

[Further background information about the topics covered in this section (2.1) may be found in bibliographic citations 2, 7, 8, 10, 18, 40, and 57.]

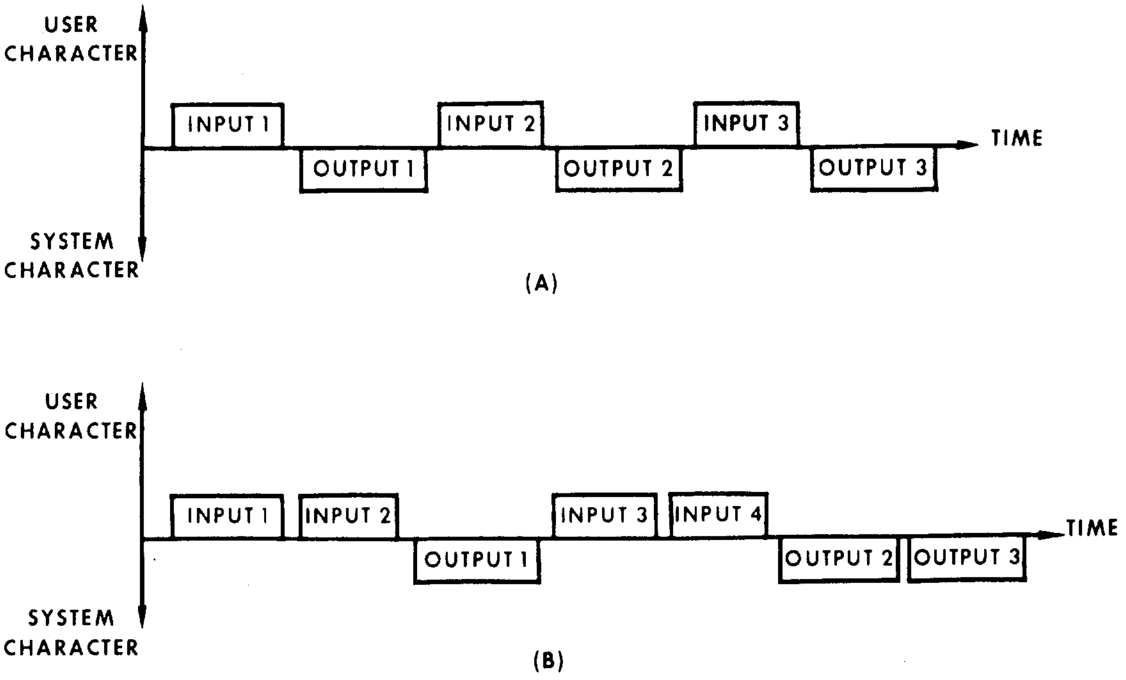

Figure 2. *Overlap of Requests and Responses*
(A) No type-ahead
(B) Type-ahead

2.2 Interactive turnaround time

From an organizational point of view, the most significant measure of interactive computer service is the amount of time required to perform a specified amount of work, known as the interactive turnaround time. Specification and measurement of turnaround time must include a description of the workload.

Interactive turnaround time is measured as the elapsed time required to complete a given (sequence of) task(s) in interactive mode.

Depending on the user's application, the meaningful turnaround time may range from a single input-output message pair (see Fig. 3A, p. 14) to the elapsed time from the beginning of the first input to the end of the last output (see Fig. 3B, p. 14). When this latter set is identified as a job, then the turnaround time is also known as the job run time.

Turnaround time includes user think time (also called delay or wait time), user transmission time, response time, and system transmission time (see Fig. 3C, p. 14). Design of the measurement activity must provide for possible variability in the user time components think time and transmission time. One objective of environmental control, discussed in section 3, is to ensure that user time does not introduce uncertainty into the measurements.

[Further background information about the topics covered in this section (2.2) may be found in bibliographic citations 3, 23, and 34.]

2.3 Applicability of response time and turnaround time

Response time calculation requires a considerable volume of relatively high precision measurements. The alternative is to measure the elapsed time to complete a given task. This involves fewer measurements over a longer interval. Therefore, less accurate measurement capabilities may be acceptable. In general, these less sophisticated measurements should be less expensive to implement. A combination of response time and turnaround time may best describe agency requirements, making it possible to express different levels of concern. Determination of both response and turnaround times require the acquisition of enough data to be statistically significant.

Comparison and evaluation of services provided by different operating systems are complicated by variance among the implementations of user aids for interactive operation. Operations accomplished in one step on some systems could require multiple steps on others. The turnaround time for the operation is the recommended service measure in this situation.

Type-ahead has less effect on the measurement of turnaround time than on response time. Among turnaround time measures, job run time is least affected by type-ahead. But, as shown in Figure 4, when type-ahead is present job run time may be decreased by (at least) the overlap of input and processing. It is therefore recommended that specifications include a statement of whether or not type-ahead is to be permitted, and, if it is permitted, what effect it has on the definitions and measures.

3. Conditions of measurement

The current state-of-the-art in the evaluation of computer service makes it necessary to do comparative rather than absolute measurement. The available measures of interactive computer service are not absolute; they are relative to the test methodology and workload. It is always necessary to qualify measurements by a description of the measurement environment and methodology. Specification and evaluation documents should contain, explicitly or by reference, the conditions, methodology, and evaluation criteria which are to be used, as well as the statistical treatment of the measured data. If an agency plans to continue measurement after a service is in use, conditions should be controlled in the same manner as during selection.

Measurement conditions can range from totally controlled to totally uncontrolled. Uncontrolled conditions imply measuring a sample of the user/system interactions of the actual user population. When conditions are uncontrolled there are two major independent variables: the user being measured and the total workload on the computer. These measurements are generally not repeatable, but they do offer insight into the behavior of the users as well as the computer. Varying degrees of control may be employed during measurement; for example, human operators may be used following a set of written

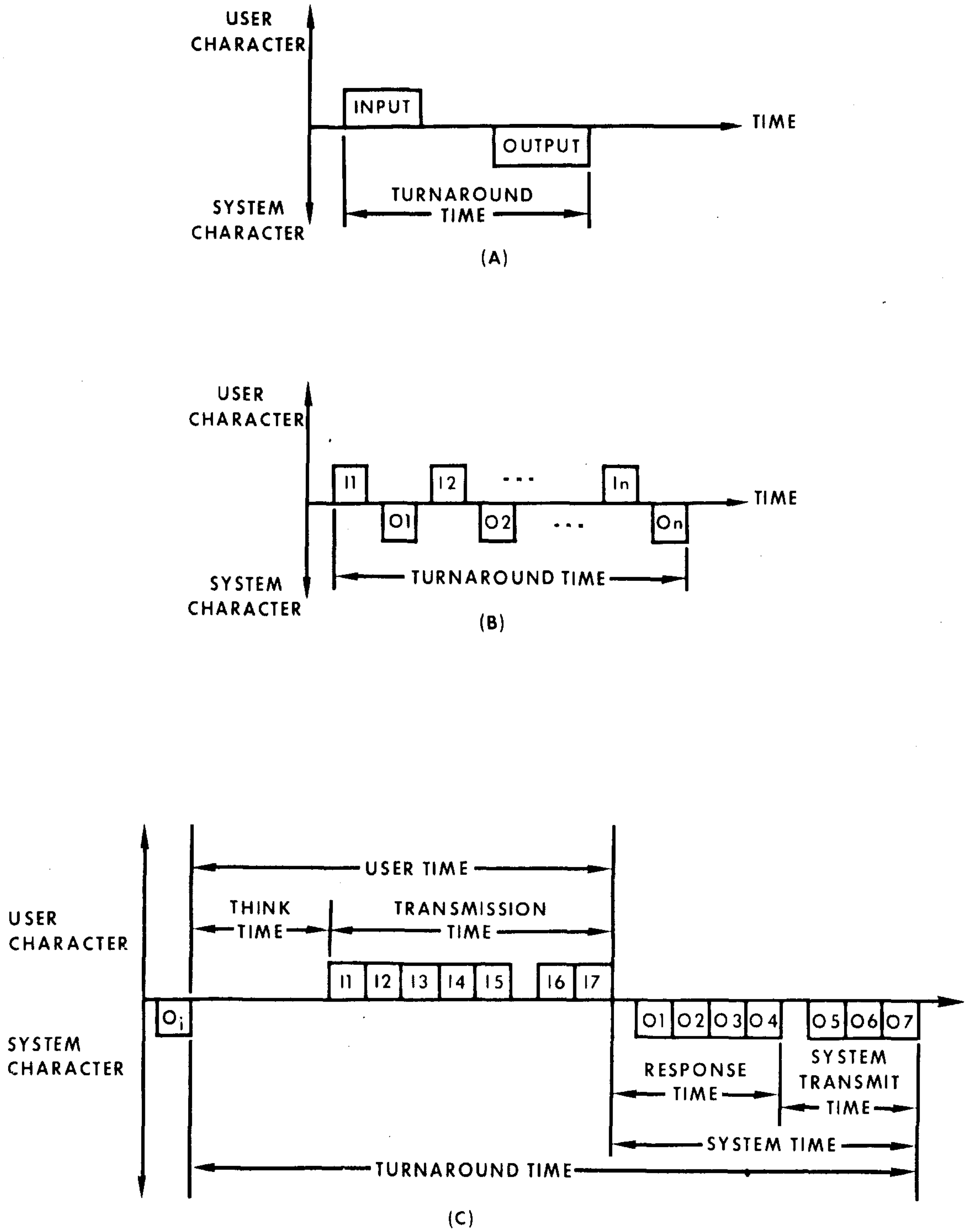

Figure 3. *Turnaround Time*
(A) Single input–output pair
(B) First input to last output (job run time)
(C) Components of turnaround time

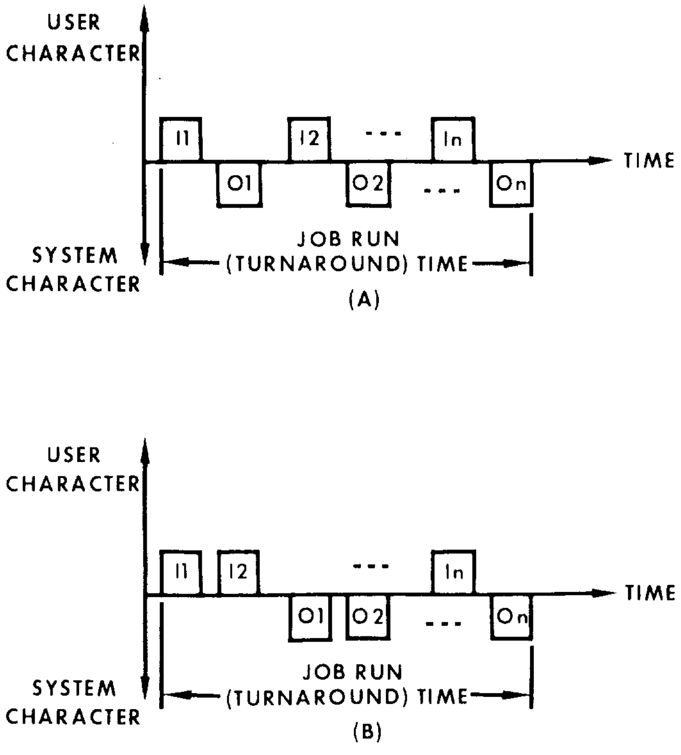

Figure 4. *Effect of Type-ahead on Job Run (Turnaround) Time*
(A) No type-ahead
(B) Type-ahead permitted

instructions, or some mechanism might be used to perform user functions automatically.

The validity of conclusions drawn from any measurement methodology critically depends on the workload. In any performance evaluation effort an essential consideration is the accuracy with which the workload applied during the evaluation represents the workload about which inferences are drawn. Controlled conditions are required for repeatable tests. It is possible that no matter how well controlled a series of tests, results may not be perfectly repeatable as a consequence of internal variations in multi-programming operating systems as well as queuing and switching delays in the communications network. Such variation is normal and should be expected. Data obtained from controlled condition measurements may be useful in providing insight to the interpretation and projection of uncontrolled behavior.

Complete control of the environment is required for such applications as stress testing, tuning, and complete systems comparison. Since these applications are beyond the scope of this document, testing with complete control will not be discussed.

[Further background information about the topics covered in this section (3) may be found in bibliographic citations 5, 7, 27, 49.]

3.1 Actual production use—no control of the environment

Actual production use of a computer service may be measured to determine the characteristic use of the service for the purpose of generating test workloads. Measurement of actual production service utilization may sometimes be substituted for controlled testing. It may be easier to collect data from actual usage than it is to design the benchmark test workload for a controlled test, but this is counterbalanced by the difficulty of obtaining useful statistics from

the potentially large volume of data collected.

The duration of the user's activities will be one of the descriptive variables when uncontrolled usage is measured. Because of user variability, turnaround time is not recommended as a service measure of actual uncontrolled usage.

The recording of users' data communications also raises several concerns relative to assuring adequate confidentiality in the acquisition, storage, analysis, and reporting of such data.

One way to measure uncontrolled usage of interactive computer service is to employ a suitable communications monitor. In order to measure the service at the delivery point, the communications monitor would have to be connected at the interface between the user's terminal and the network. Note that there must be data analysis and report generation programs to process the data obtained by the communications monitor. While it may be a formidable task to acquire the data, the design and implementation of the analysis programs may be an even larger undertaking. (See also section 4.1.3).

Whenever user-system communication is monitored, the confidentiality of data such as passwords and other restricted identifiers must be maintained. Furthermore, the content of the user's work may also be sensitive; e.g., the user may have been accessing a personnel file. The monitoring of communications should be a prudent undertaking. It may be subject to Federal and/or State regulations. Notification to the parties involved may be required. Proper procedures and safeguards must be utilized to assure that the regulations are satisfied. When establishing a procedure for monitoring, appropriate counsel should be sought.

[Further background information about the topics covered in this section (3.1) may be found in bibliographic citations 9, 13, 16, 21, 45, 52, and 57.]

3.2 Partial control of the environment

Since testing with partial control of the environment is easier, potentially less disruptive to normal operations, and less costly than testing with complete control, its use is recommended whenever the results obtainable will satisfy the objectives of the measurement program. The major objective in controlling user input is to eliminate uncontrolled variability in the time spent in the user state. Among the applications of partial control are functional demonstrations, quality control and improvement of currently used services, and comparison of alternate sources of computer services.

A functional demonstration is intended to demonstrate capabilities in some specific area without regard to total performance. Under most circumstances it should be possible to perform functional demonstrations without completely usurping the computer network. As a matter of fact, it may be part of such demonstrations to show that the remainder of the operation is not affected in any way obvious to the casual observer.

Quality assurance testing, as used in this document, is concerned with detecting and quantifying the effect of a change. This change could be in hardware or software; in the case of a computer service, it could also be in the ambient workload. For example, quality control testing is involved in the decision whether or not to install a new software release; or whether a local modification to the operating system results in an improvement or degradation in the service; or whether the service being obtained under a service contract has altered because of some change beyond the users control, such as modifications to the hardware, software, or workload.

In the context of this document, quality assurance is exercised to ensure that the level of service does not deteriorate. It is accomplished by repetition of benchmark tests after a system change. In summary, the approach is to compare the results obtained from running the benchmark test at any time with the previous history from the same test.

One very common mode for evaluating interactive computer service is to measure the service delivered to one user's terminal, with the remainder of the workload being regarded as beyond the user's control and, therefore, of no interest. This mode is consistent with the individual user's viewpoint and is relatively straightforward to implement. Partial control of the environment exists when the workload entered through this one terminal is controlled for purposes of measurement and evaluation.

[Further background information about the topics covered in this section (3.2) may be found in bibliographic citations 3, 27, 35, 38, 54, and 57.]

4. Methods of measurement and analysis

Selection of the measurement methodology and methods of data analysis must reflect both the measurement conditions and the functional performance measures. The measurement methodology selection is part of the experimental design. Results obtained employing one methodology may not be compatible with those obtained when another methodology is employed. Cost, complexity, and accuracy must weigh in the selection. Associated with every component of the measurement methodology must be a procedure for analyzing acquired data. Both data acquisition and analysis must be verifiably accurate and precise within specified limits. Validation techniques must be applied to the selected methodology to insure that it is properly applied and that it functions properly during use. It is also necessary to confirm that the selected methodology or implementation is properly presenting the workload specified. (Note that workload specification is a difficult and important topic beyond the scope of these guidelines.)

Repeated testing is fundamental to measurement and analysis. When measuring a service, tests should be conducted periodically to determine if the quality of service has changed; as well as before and after a known change in the service's operating system, applications programs, computer hardware configuration or communication facility configuration. During selection, the test results often constitute one of the major factors in determining the acceptability of a given service. Test design must reflect the accuracy with which data is recorded and the analysis to which the data will be subjected.

The following list of methodologies is arranged in order of increasing complexity and cost. Although functional performance measures may eliminate the need for the simpler ones in some applications, they are included for completeness. The first three (accounting log, stopwatch, and communications monitor) are purely data acquisition devices; the next two (users at terminals) and automatic send-receive terminals are purely drivers—means of introducing a test workload; the final three (intelligent terminals, internal drivers, and remote terminal emulators) may combine functions of drivers and monitors.

Selection of the measurement methodology is part of the design of the measurement program. The selection must include considerations of the purpose for which measurement is being conducted, the conditions of measurement, the requirements for repeatability, the number of interactive terminals to be measured, and the cost of performing the measurements. It is, therefore, impossible to select a preferred methodology *in vacuo*. Since the following list of methodologies is arranged in order of increasing complexity and cost, the methodology which occurs first in the list which meets all the boundary considerations is the prime candidate for application.

[Further background information about the topics covered in this section (4) may be found in bibliographic citations 1, 5, 24, 37, and 57.]

4.1 Data acquisition devices

4.1.1 The accounting log

Virtually all medium- and large-scale computer systems are provided with "accounting programs" to record selected events occurring during operation. It may have the capability to analyze the data necessary to determine response time and many other useful measures of system utilization and performance. Many computer networks are likewise provided with accounting programs which measure the utilization and effectiveness of the communications system. Such techniques provide information only on the use of the computer system employed to produce the service and, separately, the communication facility employed as the delivery mechanism. It is presently difficult, if not impossible, to combine these data to obtain information concerning the quality of the service as seen by the end user.

Since the data recorded in the accounting log and the analysis produced by the accounting program is specific to a particular operating system and applications program, this methodology's usefulness is restricted to comparison of homogeneous systems or testing of a single system. Only if it could be proved that data obtained from heterogeneous systems was in fact comparable could this data be used for inter-system comparison and evaluation. For the evaluation of the service delivered to an individual user, the data may not be available from all servers; even if available, they may not be comparable.

Accounting data is obtained at some point internal to the computer system providing the service; it excludes the time required for input and output to travel between that point and the user's terminal. This internal point of measurement makes the data incompatible with the definitions of response and turnaround times. Accounting data cannot be recommended for comparison of services. Its use in system evaluation should be examined closely in consideration of the preceding limitations.

[Further background information about the topics covered in this section (4.1.1) may be found in bibliographic citations 5, 7, 15, 17, 18, and 59.]

4.1.2 Stopwatch

The simplest mechanism for measuring computer service performance is a stopwatch. Since data recording is subject to human error as well as normal inaccuracies due to human response time, measurement of relatively long elapsed times (such as those related to turnaround time as compared to response time) are recommended, as is an agreed upon method for resolving conflicts among differing data recorded by independent observers.

The use of stopwatches is most common when the input is controlled. In general, the accuracy of stopwatch data is increased when the test is rehearsed or repeated. Analysis of stopwatch data may be performed manually or by computer; in either case provision must be made for prevention and detection of human transcription errors.

[Further background information about the topics covered in this section (4.1.2) may be found in bibliographic citations 7 and 49.]

4.1.3 Communications monitor

A communications monitor may be employed to collect data about response time and turnaround time by connection to the communication interface(s) of the terminals or access ports to the network service. Since these interfaces are designed to connect to communications equipment, such connection is relatively easy and unlikely to perturb the equipment to which connection is made.

The interface between data terminal equipment (DTE) such as terminals and computers, and data circuit-terminating equipment (DCE), such as modems, multiplexors, and network access ports, is the object of several standards and proposed standards. For the purposes of this document, it is sufficient to note that these interface standards define a convenient and well defined point to connect a communications monitor with minimum potential of affecting the communication over the data link.

Communications monitors are receive-only DTE which connect to the data communications circuit at the interface to the DCE (or at an electrically or logically equivalent point) in parallel with the pre-existing equipment. This connection is accomplished by a T-connection, as illustrated in Figure 5.

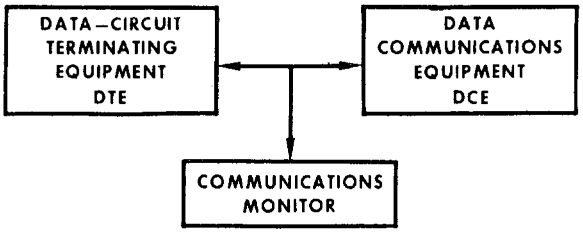
DATA-CIRCUIT TERMINATING EQUIPMENT DTE
DATA COMMUNICATIONS EQUIPMENT DCE
COMMUNICATIONS MONITOR

Figure 5. *Connection of Communications Monitor*

Monitors could be specially designed for communications performance testing, or could be general purpose hardware monitors augmented with a small amount of communications-oriented hardware. To be useful in this environment, a monitor would have to provide a means for timing the communications traffic, and would have to provide for recording sufficient volumes of both traffic and timing. Communications monitors are useful under both controlled and uncontrolled conditions.

[Further background information about the topics covered in this section (4.1.3) may be found in bibliographic citations 2, 3, 41, 42, and 45.]

4.2 Test drivers

4.2.1 User(s) at terminals

As an alternative to the measurement of uncontrolled usage, operators at terminals may be employed in an attempt to impose a controlled workload under limited situations. However, since terminal operators' performance is unrepeatable, error prone, and introduces undesirable variability into the time spent in the user state, operators should not be employed for this purpose unless other more desirable alternatives are infeasible or unavailable.

[Further background information about the topics covered in this sections (4.2.1) may be found in bibliographic citations 7 and 22].

4.2.2 Automatic send-receive terminals

When only a few terminal interactions need to be controlled, the use of Automatic Send-Receive (ASR) terminals to represent operator input is a very attractive and relatively simple technique. The operator input is stored on some medium such as paper or magnetic tape or some other memory associated with the terminal, depending on how the terminal is equipped. The terminal is operated with the stored input to produce communications traffic equivalent to that which would have been produced by a user. Barring malfunction, this input is controllable and repeatable. Unless the terminal has been modified to produce automatic data collection, interactions involving ASR terminals are best timed with respect to turnaround time either manually or by obtaining the time from the computer system as part of the interaction.

Punched tape was the first storage media employed by ASR terminals. More recently, magnetic tape cassettes and cartridges have become available. The most recent advance to ASR technology has been the incorporation of memory into terminals. (When processing capability as well as memory is added, the terminal is classified as intelligent. Intelligent terminals are discussed below.)

When a terminal is operating in ASR mode, the input comes from the media rather than the keyboard. This alternate source of input introduces a control problem when used to replace the human communicating with the computer. The problem is how to stop input at the point where the human would stop and how to resume at the point where the human would start. This is a non-trivial problem in general.

A conventional solution has been to employ two of the ASCII device control characters to stop and start input. Where these controls are not available, input data may be lost. There must be a stop-code stored at the end of each user input; the terminal must usually be operating in half-duplex for any assurance that the stop-code be effective. The computer system must issue a start-code at the end of its output to start the input for the next interaction.

The workability of this approach is dependent on the cooperation of the computer service. It must accept and disregard the stop-code transmitted to it. It must also transmit the appropriate start-code when ready to accept input.

There is a flow-control problem associated with the use of ASR terminals which employs these device control characters for its solution. The problem is that the character transmission rate from the terminal operating in ASR mode is usually much higher than when input comes from a human operator. Transmission in ASR mode is usually at the rated channel capacity. Operator input has been observed to average 4-5 words per minute in certain condition up to a maximum typing rate of 60-100 words per minute. The 300 bit per second transmission channel commonly available supports 300 words per minute. Most networks cannot support extended input at this rate. Control is achieved by transmission of the stop-code to temporarily suspend input and the start-code to resume input.

The use of the device control characters for flow control as well as invitation for input, requires that the terminal and server computer network employ the same conventions. Unfortunately since standards do not exist, flow control is not universally available.

When a computer service requires explicit operator action to enable ASR mode, the use of ASR terminals to represent operator input becomes highly questionable. The explicit operator action may have alerted the service that this terminal must be treated differently from the norm. One does not know how this different treatment is implemented or how it affects the service rendered. The representativeness of the test must be examined.

4.3 Driver/monitor combinations

4.3.1 Intelligent terminals

Some of the limitations on the use of ASR terminals may be eliminated by the use of intelligent terminals. While theoretically attractive, this approach has not been given much application. The Remote Terminal Emulator discussed below embodies this concept.

The addition of processing capability to an ASR terminal can potentially eliminate the restrictions of the ASR mode. These terminals, known as intelligent terminals because of their processing capability, can execute commands stored along with the text of the interaction between user and computer.

Depending on the programming flexibility of the intelligent terminal and the regularity and predictability of the computer's output, it may be possible to emulate a person using the terminal for computer communication to the extent that from the computer's end of the communications channel it is impossible to distinguish between actual usage on a test. If the intelligent terminal were also to function as a monitor, a high resolution clock and a data recording medium would have to be included in the configuration. When this state has been achieved, the intelligent terminal has become a single–user Remote Terminal Emulator (RTE). RTE's are discussed in detail in section 4.3.3.

4.3.2 Internal measurement drivers

Although the use of a driver internal to a computer is not recommended for evaluation and selection of network service provided in part by that computer, it may be useful for such purposes as: i) comparison of the computer system (host computer) components of network service, which are completely homogeneous in hardware and software, ii) testing for compliance with standards (such as language and communications protocol), iii) tuning of the computer, and iv) detecting the effect of change in host hardware, software, or utilization. The use of an internal driver is less expensive and complicated than an external driver since only one computer is required.

The teleprocessing workload may be emulated by a program running internal to a network host computer, either in the central processing unit, the communications front–end, or, when the architecture supports it, some other processor configured as part of the system. These programs are known collectively as internal drivers or internal stimulators. The monitor function is included in the internal driver explicitly or by use of the accounting log since there might be no external communication. These internal drivers range in sophistication from ones which simply read a simulated terminal communication from a storage device such as tape or disk and present it to the operating system or applications program, to ones which incorporate all the complexity of the Remote Terminal Emulators (RTE's) (discussed in the following section) including the use of a dedicated communications processor which is externally cabled to the communications device which would normally be configured. Most of the software systems used to implement Remote Terminal Emulators can also operate as internal drivers.

With an internal driver the communications handling of the host may be bypassed. When the driver is executed in some processor other than the CPU, the communications handling of the host may be exercised to some degree; however, the procedures are not the same since the interrupts are being generated internal to the host. Since many internal drivers can bypass various amounts of hardware and software, depending on how the system software is generated, it is extremely difficult to establish exactly what is being tested.

An internal driver, at best, tests the service at the communications interface to the host, not at the user's terminal. Since the service delivered at the user's terminal includes the effect of the communications network as well as the computer system, internal drivers are inadequate for testing computer service.

[Further background information about the topics covered in this section (4.3.2) may be found in bibliographic citations 1, 46, and 58.]

4.3.3 Remote terminal emulators

When an external computer is used to provide the workload on an interactive computer network service, the computer performing the testing is known as a Remote Terminal Emulator (RTE) and the entire network being tested is the Service Under

Test (SUT). The capacity of RTE's can range from one terminal to the maximum number which the service can support. An RTE may be an appropriate test tool when there is a specified number of access ports which the service must provide. (When only one port is provided, a single-user RTE implemented in an intelligent terminal may be appropriate.)

While the statistical treatment of data is important in all measurement and evaluation endeavors, the volume of data which may be collected by an RTE makes necessary the complete specification of what data is to be recorded, how the data analysis and report generation is to occur, and how the correctness of the data reduction software is to be established.

Remote terminal emulation is an approach to the performance evaluation of teleprocessing services in which a driver external to and independent of the Service 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 it were a set of terminal devices and operators. A Remote Terminal Emulator (RTE) is a specific implementation of a teleprocessing workload driver employed in remote terminal emulation. 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. Performance determinations are made through subsequent analysis of this data.

When a mechanism as complicated as a Remote Terminal Emulator is used, there are many opportunities for performance of the RTE itself to deviate from specifications. In addition to hardware malfunctions, the possibility exists for software or manual errors in the translation of the specified workload into a form which the RTE uses to produce the actual test workload. It is therefore necessary to verify that the workload being imposed is that which was specified. Among the techniques applicable to this verification are inspection of audit trail recordings of all communication between the RTE and SUT, usually maintained by the RTE, and the accounting records of the SUT. The "broadcasting" of a message by the operator of the SUT to all terminals, and the interrogation of the time-of-day clock by the (emulated) user are additional recommended techniques when such capabilities are available.

Remote terminal emulators range widely in size and capability. Emulating a single user can be a simple task performed by an intelligent terminal. Emulating hundreds of terminals, and checking the correctness of the responses received at each one, may require the services of a large computer system. While the latter may be appropriate for testing an entire system, more modest implementations will suffice for service evaluations; minicomputer-based RTE's should be considered.

[Further background information about the topics covered in this section (4.3.3) may be found in bibliographic citations 1 and 58.]

4.4 Data analysis and presentation

The data collected by the selected measurement method is analyzed and employed as the basis for reports. The most stringent requirements on these analysis and report generation programs occur when two or more services are being compared.

Measurement of response time requires the accumulation of large volumes of data, which must be described by the use of statistics. Measurement of turnaround time results in a smaller volume of data, but one which is still potentially unmanageable except statistically. The statistical techniques employed can influence the results; therefore, the techniques must be clearly specified in advance. The sampling method, standard sampling interval, and data grouping are among the techniques which must be specified. Due to the observed naturé of the data distributions, use of the median and other percentile statistics is preferable.

The analysis and presentation of data describing interactive computer service is independent of the method employed to acquire the data. Therefore, verification of the data analysis may be considered independently. Basically, there are three ways in which the validation and verification can be performed.

Given the acquired data, the required calculations can be performed manually to produce the specified measures. Then these manually-acquired results are compared with the results of the machine-implemented routines. If calculations are to be performed manually, it is necessary to have substantial redundancy in order to decrease the probability of error.

The second verification procedure involves independent collection and/or analysis of the data describing the service which is compared with the results from the analysis routines associated with the driver. It may be done with the aid of an independent hardware monitor which is used in addition to whatever monitor is normally used by the driver technique, or the data may be processed by a separate analysis program. If computerized analysis routines are employed, they must be tested to verify their accuracy.

The third procedure requires that the analysis routines be applied to known data. This known data is generated to conform to a given statistical distribution. This data is input to the analysis routines and the results compared with the known distribution.

Quite often the distributions of response time and turnaround time are not known (in closed mathematical form). Since most of the data distributions which have been collected and analyzed have been non-normal (non-Gaussian), Gaussian statistical descriptors such as mean and standard deviation are not appropriate. It is therefore recommended that non-parametric statistical descriptors be employed when specifying response time and turnaround time. Percentile statistics such as the median, 90% level, and 95% level are most commonly used.

[Further background information about the topics covered in this section (4.4) may be found in bibliographic citations 2, 22, and 57.]