American National Standard

for information systems –

ADOPTED FOR USE BY THE FEDERAL GOVERNMENT

FIPS

PUB 155

SEE NOTICE ON INSIDE

data communication systems and services – measurement methods for user-oriented performance evaluation

ANSI X3.141-1987

american national standards institute, inc.
1430 broadway, new york, new york 10018

This standard has been adopted for Federal Government use.

Details concerning its use within the Federal Government are contained in Federal Information Processing Standards Publication 155, Data Communication Systems and Services User-Oriented Performance Measurement Methods. For a complete list of the publications available in the Federal Information Processing Standards Series, write to the Standards Processing Coordinator (ADP), Institute of Standards and Technology, Gaithersburg, MD 20899.

ANSI®
X3.141-1987

American National Standard
for Information Systems –

Data Communication Systems and Services – Measurement Methods for User-Oriented Performance Evaluation

Secretariat

Computer and Business Equipment Manufacturers Association

Approved October 8, 1986

American National Standards Institute, Inc

American National Standard

Approval of an American National Standard requires verification by ANSI that the requirements for due process, consensus, and other criteria for approval have been met by the standards developer.

Consensus is established when, in the judgment of the ANSI Board of Standards Review, substantial agreement has been reached by directly and materially affected interests. Substantial agreement means much more than a simple majority, but not necessarily unanimity. Consensus requires that all views and objections be considered, and that a concerted effort be made toward their resolution.

The use of American National Standards is completely voluntary; their existence does not in any respect preclude anyone, whether he has approved the standards or not, from manufacturing, marketing, purchasing, or using products, processes, or procedures not conforming to the standards.

The American National Standards Institute does not develop standards and will in no circumstances give an interpretation of any American National Standard. Moreover, no person shall have the right or authority to issue an interpretation of an American National Standard in the name of the American National Standards Institute. Requests for interpretations should be addressed to the secretariat or sponsor whose name appears on the title page of this standard.

CAUTION NOTICE: This American National Standard may be revised or withdrawn at any time. The procedures of the American National Standards Institute require that action be taken to reaffirm, revise, or withdraw this standard no later than five years from the date of approval. Purchasers of American National Standards may receive current information on all standards by calling or writing the American National Standards Institute.

Published by

American National Standards Institute
1430 Broadway, New York, New York 10018

Printed in the United States of America

PC1M589/20

Foreword

(This Foreword is not part of American National Standard X3.141-1987.)

This standard is the second of two related American National Standards that jointly provide a uniform means of describing the performance of data communication services from the point of view of data communications users. The first standard, American National Standard for Information Systems - Data Communication Systems and Services - User-Oriented Performance Parameters (ANSI X3.102-1983), defines 21 data communication performance parameters. Each parameter provides a means of specifying a particular aspect of data communication system performance in quantitative terms, from the end user's viewpoint. The parameters focus on the performance provided to pairs of individual users, but they may also be applied, with appropriate specification of usage conditions, to characterize the overall performance of data communication systems serving many users. The parameters are applicable to all classes of data communication systems, independent of topology, protocol, code, or other design characteristics. Although the parameters were chosen to describe performance as observed at the end user interfaces, they may also be used to describe the performance of digital subsystems.

This second standard specifies uniform methods of measuring values for the parameters described in ANSI X3.102-1983. The measurement methods in this standard are general and implementation independent. They may be used in measuring performance values at any pair of digital interfaces connecting a data communication system or subsystem to its users. They may be used in achieving a wide variety of measurement objectives including service or network performance characterization, comparison of observed performance values with specifications, and the determination of system design, operation, or usage effects. They may be used in comprehensive experiments in which values for all standard parameters are determined or, with appropriate simplification, in very selective experiments -- for example, the measurement of a single parameter such as Bit Error Probability.

To permit users maximum flexibility in implementation, this standard imposes measurement requirements only to the extent necessary to ensure the validity, comparability, and proper interpretation of experiment results. Particular measurements may be implemented in a variety of ways -- for example, using either two-point or loopback techniques. To clarify and facilitate its use, the standard provides references and an Appendix that describe typical implementations.

Suggestions for improvement of this standard will be welcome. They should be sent to the Computer and Business Equipment Manufacturers Association, 311 First Street NW, Suite 500, Washington, DC 20036.

This standard was processed and approved for submittal to ANSI by the Accredited Standards Committee on Information Processing Systems, X3. Committee approval of the standard does not necessarily imply that all members voted for its approval. At the time it approved this standard, the X3 Committee had the following members:

Organization Represented	*Name of Representative*
AMP Incorporated	Patrick E. Lannan Edward Kelly (Alt)
Association for Computing Machinery	Kenneth Magel Jon A. Meads (Alt)
Association of the Institute for Certification of Computer Professionals	Thomas M. Kurihara
AT&T Communications	Henry L. Marchese Richard Gibson (Alt)
AT&T Technologies	Herbert V. Bertine Paul D. Bartoli (Alt) Stuart M. Garland (Alt)
Burroughs Corporation	Stanley Fenner
Control Data Corporation	Charles E. Cooper Keith Lucke (Alt)
Cooperating Users of Burroughs Equipment	Thomas Easterday Donald Miller (Alt)
Data General Corporation	John Pilat Lyman Chapin (Alt)
Data Processing Management Association	Christian G. Meyer Ward Arrington (Alt) Terrance H. Felker (Alt)
Digital Equipment Computer Users Society	William Hancock Dennis Perry (Alt)
Digital Equipment Corporation	Gary S. Robinson Delbert L. Shoemaker (Alt)
Eastman Kodak	Gary Haines Charleton C. Bard (Alt)
General Electric Company	Richard W. Signor William R. Kruesi (Alt)
General Services Administration	William C. Rinehuls Larry L. Jackson (Alt)
GUIDE International	Frank Kirshenbaum Sandra Swartz Abraham (Alt)
Harris Corporation	Walter G. Fredrickson Rajiv Sinha (Alt)
Hewlett-Packard	Donald C. Loughry
Honeywell Information Systems	Thomas J. McNamara David M. Taylor (Alt)
IBM Corporation	Mary Anne Gray Robert H. Follett (Alt)
IEEE Computer Society	Sava I. Sherr Thomas M. Kurihara (Alt) Thomas A. Varetoni (Alt)
Lawrence Berkeley Laboratory	David R. Stevens Robert L. Fink (Alt)
Moore Business Forms	Delmer H. Oddy
National Bureau of Standards	Robert E. Rountree James H. Burrows (Alt)
National Communications System	George W. White
NCR Corporation	Thomas W. Kern A. Raymond Daniels (Alt)
Prime Computer, Inc	Arthur Norton Linda J. Allen (Alt)
Railinc Corporation	R. A. Petrash
Recognition Technology Users Association	Herbert F. Schantz G. W. Wetzel (Alt)
Scientific Computer Systems Corporation	James A. Baker Carl Haberland (Alt)
SHARE, Inc	Thomas B. Steel Robert A. Rannie (Alt)
Sperry Corporation	Marvin W. Bass Jean G. Smith (Alt)
Texas Instruments, Inc	Presley Smith Richard F. Trow, Jr (Alt)
3M Company	Paul D. Jahnke J. Wade Van Valkenburg (Alt)
Travelers Insurance Companies, Inc	Joseph T. Brophy
U.S. Department of Defense	Fred Virtue Belkis Leong-Hong (Alt)

Organization Represented	*Name of Representative*
VIM	Chris Tanner Madeline Sparks (Alt)
VISA U.S.A.	Jean T. McKenna Susan Crawford (Alt)
Wang Laboratories, Inc	Marsha Hayek Joseph St. Amand (Alt)
Xerox Corporation	John L. Wheeler Roy Pierce (Alt)

Subcommittee X3S3 on Data Communication, which developed this standard, had the following members:

W. F. Emmons, Chair

H. V. Bertine
P. Campbell
W. S. Carter
J. W. Conway
H. S. Crowley
K. L. Dally
S. DiCecco
M. Griefner
E. R. Hansson
J. T. LaBanca
G. Lawrence
F. Piper
J. Quale
C. Venditto
M. Weiss
J. L. Wheeler
G. W. White

Task Group X3S3.5 on Quality of Service, which had technical responsibility for the development of this standard, had the following members:

N. B. Seitz, Chair

H. Berkowitz
D. Carpenter
A. Gallo
S. Garde
S. Holmgren
N. Louthain
R. Paolucci
S. Richman
T. D. Zimmerman

Other persons who contributed to the development of this standard include the following:

D. Bodson
E. L. Crow
D. S. Grubb
P. M. McManamon
M. J. Miles
R. T. Moore
K. P. Spies
D. R. Wortendyke

Contents

SECTION PAGE

SECTION PAGE

Figures

American National Standard
for Information Systems -

Data Communication Systems and Services - Measurement Methods for User-Oriented Performance Evaluation

1. Scope and Introduction

This standard specifies uniform methods of measuring the performance of data communication services at digital interfaces between data communication systems and their users. These methods may be used to characterize the performance of any data communication service in accordance with the performance parameters defined in American National Standard for Information Systems - Data Communication Systems and Services - User-Oriented Performance Parameters, ANSI X3.102-1983 [1].[1] Perfomance measurements conducted in accordance with this standard are intended to be used in making technical or business decisions concerning the provision and use of data communication systems and services.

Table 1 lists the 21 performance parameters defined in ANSI X3.102-1983. These parameters describe the performance of three principal data communication functions: access, user information transfer, and disengagement. The parameters are of two basic types: primary parameters and ancillary parameters. The primary parameters describe performance with respect to three general concerns (or performance criteria) frequently expressed by data communications users: speed, accuracy, and reliability. The ancillary parameters describe the influence of user delays on the primary speed parameter values.

The parameters[2] are defined in such a way that they can be applied to any data communication system or service, independent of topology, protocol, code, or other design characteristics. Although the parameters were chosen to describe performance as observed at the end user interfaces, they may also be used to describe the performance of digital subsystems. The parameters focus on the performance provided to individual user pairs, but they may also be used, with appropriate specification of usage conditions, to characterize the overall performance of data communication systems serving many users. The measurement methods defined here are intended to provide that same general applicability.

Figure 1 illustrates the process of data communication performance measurement as described in this standard. Inputs to the measurement process consist of (1) measurement objectives defined by the experiment context and (2) digital signals observed at the monitored user/system interfaces. Results of the measurement process consist of (1) estimated (mean) values for performance parameters that characterize the monitored system, and (2) associated precision and variability statistics (e.g., confidence limits, histograms, and regression coefficients). The measurement process is accomplished in four primary phases:

(1) *Experiment Design.* General measurement objectives are developed into a detailed experiment plan that defines the specific

---

[1]Numbers in brackets refer to corresponding numbers in Section 7, "References."

[2]Throughout this standard, the term "parameters" refers to the performance parameters defined in ANSI X3.102-1983.

Table 1
Matrix Representation of the Parameters

FUNCTION	PERFORMANCE CRITERION: SPEED	PERFORMANCE CRITERION: ACCURACY	PERFORMANCE CRITERION: RELIABILITY	PERFORMANCE TIME ALLOCATION
ACCESS	ACCESS TIME	INCORRECT ACCESS PROBABILITY	ACCESS DENIAL PROBABILITY ACCESS OUTAGE PROBABILITY	USER FRACTION OF ACCESS TIME
USER INFORMATION TRANSFER	BLOCK TRANSFER TIME	BIT ERROR PROBABILITY BIT MISDELIVERY PROBABILITY EXTRA BIT PROBABILITY BLOCK ERROR PROBABILITY BLOCK MISDELIVERY PROBABILITY EXTRA BLOCK PROBABILITY	BIT LOSS PROBABILITY BLOCK LOSS PROBABILITY	USER FRACTION OF BLOCK TRANSFER TIME
	USER INFORMATION BIT TRANSFER RATE	TRANSFER DENIAL PROBABILITY		USER FRACTION OF INPUT/OUTPUT TIME
DISENGAGEMENT	DISENGAGEMENT TIME	DISENGAGEMENT DENIAL PROBABILITY		USER FRACTION OF DISENGAGEMENT TIME

Legend:
- Primary Parameters (unshaded)
- Ancillary Parameters (hatched)

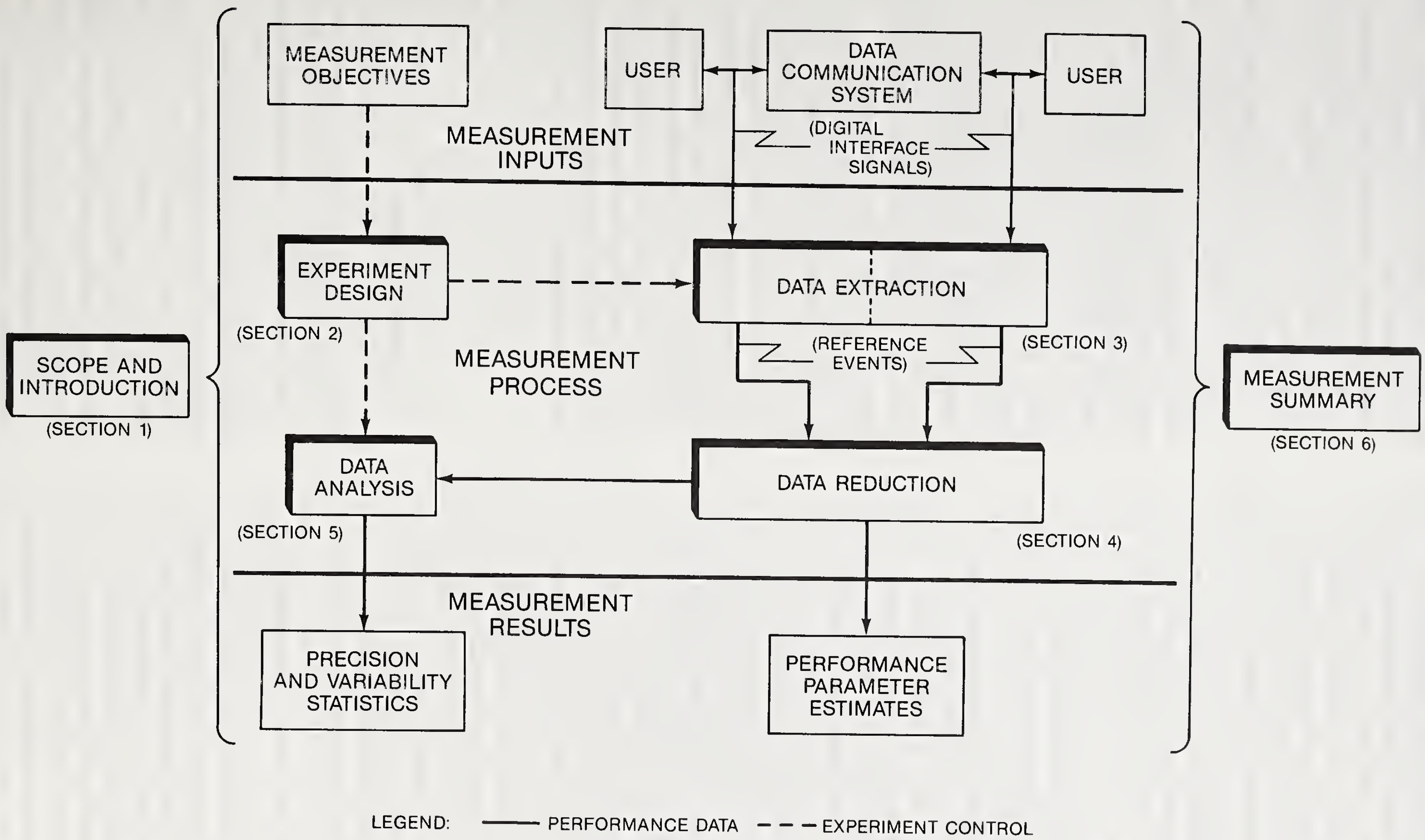


Figure 1
Overview of the Standard

performance information to be collected and the focus and conditions of individual tests.

(2) *Data Extraction*. Signals transferred across selected pairs of digital user/system interfaces are monitored in real time. At each monitored interface, the nature and time of occurrence of performance-significant interface signals are recorded in a chronological reference event history.

(3) *Data Reduction*. The recorded reference event histories are merged and processed to produce estimated values for selected performance parameters.

(4) *Data Analysis*. The reduced data are examined statistically to determine the precision of individual parameter estimates and any associated conclusions.

Figure 1 also illustrates the organization of this standard. Section 2 defines a general procedure for the design of data communication performance measurement experiments. Sections 3 and 4 specify functional requirements for data extraction and data reduction systems, respectively. Section 5 outlines methods of analyzing measured performance data and defines statistical information that should be reported with the estimated means. Section 6 provides standard forms that may be used in summarizing such results. Selected references are provided in Section 7.

Two appendixes are also provided. Appendix A is a glossary of specialized data communication performance assessment terms used in this standard and ANSI X3.102-1983. Appendix B describes a hypothetical performance measurement experiment that illustrates how this standard might be applied in actual performance measurements.

The measurement methods specified in this standard focus on performance assessment and do not address other important characteristics of data communication service, such as functionality and cost. These characteristics should be separately assessed in a comprehensive data communication service evaluation.

For completeness, the measurement methods specified here address the most stringent type of application, in which all 21 performance parameters are to be measured via continuous, independent observation of events at two or more physically distant interfaces during a performance measurement period. Particular applications may be much simpler. Measurements may be restricted to a subset of the parameters, or even a single parameter. Interface observations may be limited to events of interest, and may be intermittent to minimize measurement overhead. Monitored interfaces may be located in the same physical facility, or even the same equipment, to permit the use of loopback techniques.

The measurement requirements specified in this standard are general and implementation independent. The experiment design and data analysis requirements may be implemented using a variety of statistical methods (e.g., Latin square and balanced block designs, analysis of variance, linear regression) described in the referenced literature. Data extraction can be accomplished by test sets attached to physical network interfaces, by measurement software embedded in data terminals or network nodes, or by hybrid (hardware/software) monitors. Test data may be extracted at any digital interface, and the separate interfaces monitored during a particular test need not be physically or functionally identical. Either active "event generators" or passive "event monitors" may be used. Data reduction can be accomplished either by off-line processing of recorded event histories or by on-line updating of counters and timers. In each section of the standard, measurement requirements are imposed only to the extent necessary to ensure the validity, comparability, and proper interpretation of the experiment results.

2. Experiment Design

This section defines a general procedure for the design of experiments in which estimates for performance parameters and associated precision and variability statistics are determined. Its purpose is to guide the data extraction and data analysis phases of the measurement process to achieve three general objectives:

(1) Absence of bias and explicitly stated precision in the measured values (measurement accuracy)

(2) Clearly defined applicability of the measurement results and any associated conclusions

(3) Efficient use of resources (e.g., time and cost)

The experiment design procedure consists of seven steps:

(1) *Experiment objectives*. Define the ultimate objectives of the experiment in a decision context.

(2) *Measured parameters*. Select the particular performance parameters to be measured.

(3) *Population definition*. Define the overall population of performance trials, such as access attempts, on which the measurements will focus (and to which the measured values will refer).

(4) *Performance factors*. Specify the factors (system and usage variables) presumed to influence performance; the relevant levels (or values) of each factor; and the factor combinations (groups of factor levels) of interest.

(5) *Population sample*. Select from the defined population a representative sample of performance trials to be tested.

(6) *Test conditions*. Specify a particular factor combination to be used in each test.

(7) *Mathematical model*. Where appropriate, summarize the experiment design in an explicit mathematical model.

Step (1) provides the foundation for all subsequent steps. Steps (2) through (4) define the specific performance information to be collected. Steps (5) and (6) define the focus and conditions of individual tests. Step (7) provides a concise synopsis of the design of the experiment and a basis for the data analysis. Although the indicated order is usually the most appropriate, these steps are interdependent and often must be performed iteratively in conducting an actual experiment. For example, it is frequently desirable to examine postulated performance effects qualitatively in a simple preliminary experiment before embarking on more extensive measurements.

Description of the experiment design procedure requires the definition of a number of specific measurement terms. A **trial** is an individual attempt to perform a specified data communication function: access, user information transfer, or disengagement. A **population** is a set comprising all trials of interest in a particular experiment. A population may be comprised of any one of the three types of trials. A **sample** is the subset of a population actually measured in an experiment. The relationships among these terms are illustrated in Figure 2.

A **factor** is a system or usage variable identified in a particular experiment as influencing (or possibly influencing) measured values for the performance parameters. **Levels** are the defined states or values a given factor assumes in an experiment. A **factor combination** is a set specifying a particular level for each factor of interest. A **test** is a process of data extraction that is continuous in time and involves only one factor combination. The conditions existing during a test are thus defined by a particular factor combination.

An example will help to clarify these concepts. In an experiment intended to measure block transfer time, a *trial* might be a single attempt to transfer a block of data. The *population* of interest might be the set of all block transfer attempts initiated by the users of a data communication system. The *sample* selected for actual measurement might comprise all block transfer attempts initiated by a randomly selected group of the users during a specified time period. Two variables might be identified as *factors* of interest: block size and access line speed. Each factor might assume two *levels* during the experiment: 64 and 512 bytes in the case of block size, and 1200 and 9600 bits per second in the case of line speed. Four *factor combinations* would then be identified - the four possible groupings of a specified block size with a specified line speed. The experiment would involve a number of independent *tests*, each involving the determination of block transfer outcomes under a particular block size/line speed combination. In an experiment intended to measure the user information bit transfer rate, a trial typically consists of many consecutive block transfer attempts.

Each step in the experiment design procedure is described in succession below. Detailed information relevant to particular experiment design steps is provided in ANSI X3.102-1983 and Sections 3 through 5 of this standard (as referenced below). An example of a simple experiment design is provided in Appendix B. A more comprehensive presentation of experiment design principles is provided in [2].

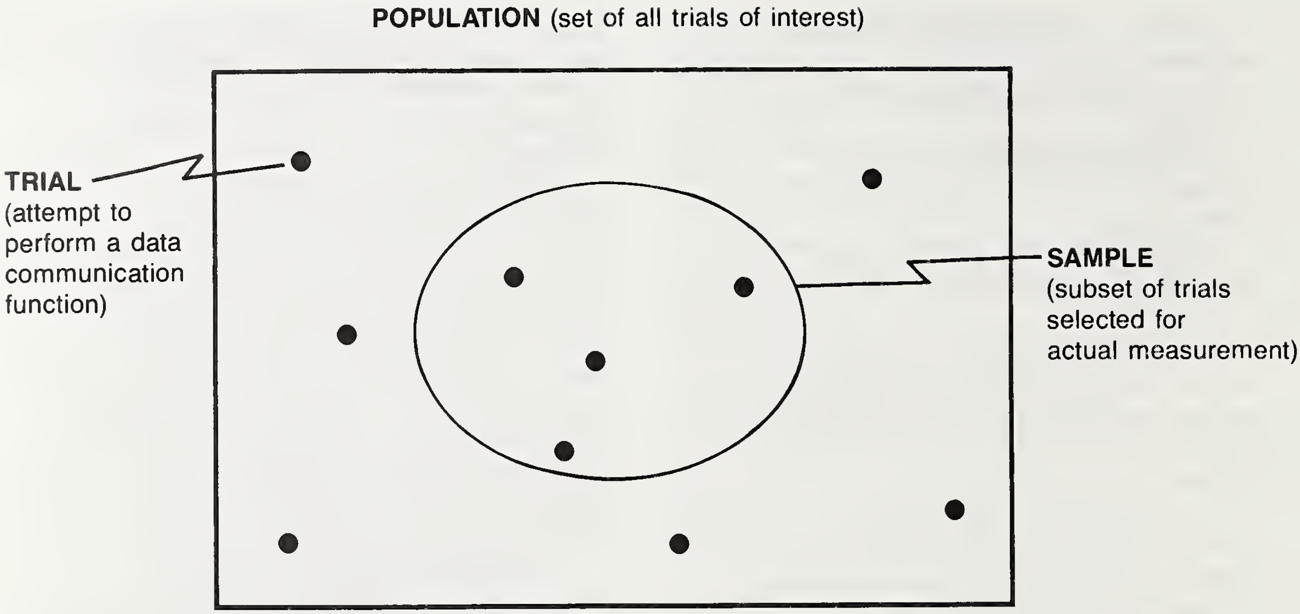


Figure 2
Relationships among Population, Trial, and Sample

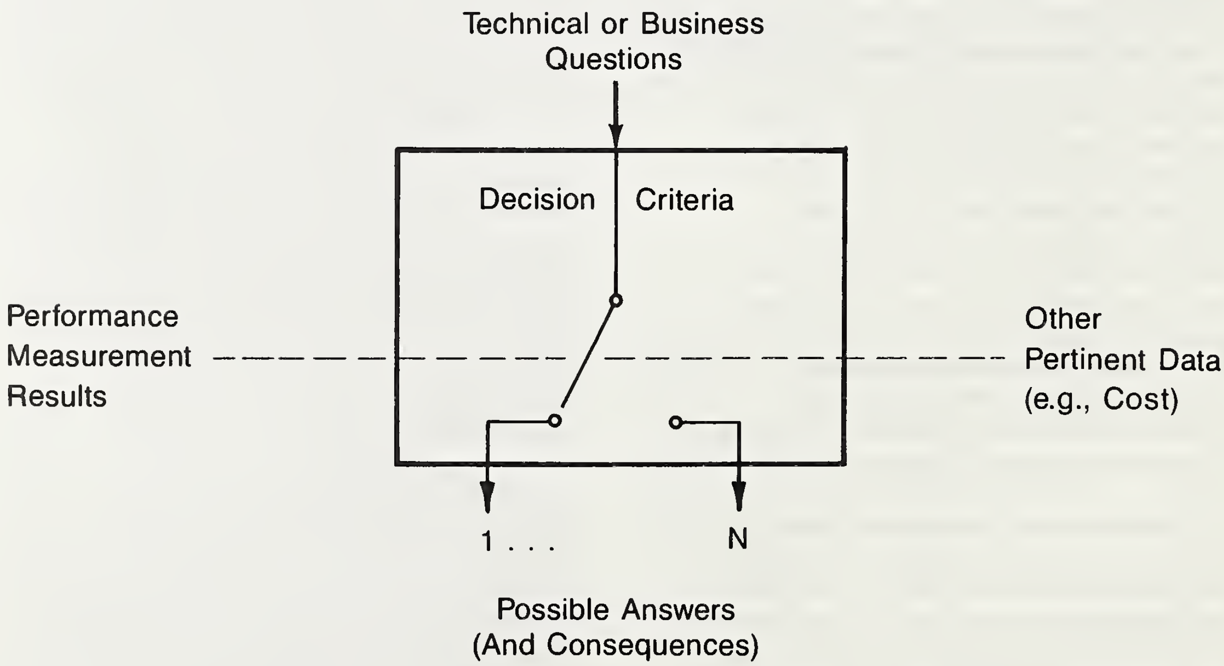


Figure 3
General Decision Context in which Performance Measurement Results May be Used

2.1 Experiment Objectives. The first step in designing a performance measurement experiment is to define the experiment objectives. This is best done by examining the decision context in which the results will be used. Three related elements of the decision context should normally be considered (Figure 3):

(1) Specific technical or business questions whose answers may be influenced by the measurement results

(2) Possible answers to these questions having substantially different practical consequences

(3) Impact of the performance measurement results (and other pertinent data such as cost) in answering each question

Examples of decisions that may be improved by performance measurement results are service selection, system design optimization, and the pricing of offered services.

Three general types of performance measurement experiments may be conducted using the guidelines presented in this standard: absolute performance characterization, hypothesis testing, and analysis of factor effects. These three experiment types involve fundamentally different experiment designs and provide substantially different information to the experimenter. They are briefly described in this subsection and are further discussed in Section 5 and the references cited there.

Absolute performance characterization experiments seek to estimate the values of selected performance parameters under a single factor combination, without reference to the effects of performance factors or previously stated performance values. Their results are most appropriately used in decision-making focused on user facilities or activities, where the communication system performance is regarded as fixed. A simulation study designed to assess the benefit of interconnecting remote computers through an existing communication system might illustrate one such use.

Simple hypothesis tests also focus on a single factor combination, but their purpose is to compare observed performance with previously stated values rather than to characterize performance in absolute terms. The results of such experiments are typically binary (e.g., delivered performance either does or does not meet a stated requirement), and they may be used directly in communication management decision-making (e.g., accept or do not accept the system). Hypothesis test results may also be used in applications such as network maintenance and control (e.g., comparing observed performance with a threshold to identify conditions requiring network reconfiguration or maintenance action).

In the third type of experiment, the analysis of factor effects, observed performance is compared among several factor combinations to identify the effects of particular performance factors. Users may apply the results of such experiments in service selection (e.g., comparing performance between providers or between the offerings of a given provider) and in service usage optimization (e.g., choosing a block length to optimize throughput). Service providers may apply results of such experiments in network design optimization (e.g., selecting among alternative routing strategies) and network management (e.g., determining traffic effects).

Performance measurements are always somewhat exploratory, and their use in communication management decision-making cannot always be foreseen. Nevertheless, all probable uses of measurement results should be considered in designing an experiment to maximize its value. The design of an experiment strongly influences the valid conclusions that may be drawn from its results.

2.2 Measured Parameters. The second step in designing a performance measurement experiment is to select the particular parameters to be measured. All or any specified subset of the parameters defined in ANSI X3.102-1983 may be measured during a particular experiment. The choice of measured parameters is determined by two factors:

(1) Experiment objectives defined in Step (1)

(2) Constraints imposed by the measurement context

It may be necessary to measure all or a large subset of the performance parameters in experiments that support service selection, acceptance testing, or the identification of long-term performance trends. Interest will often be restricted to a smaller subset of parameters in experiments that support system design and service usage optimization. The principal constraints that influence the selection of measured parameters are measurement time, available data extraction facilities, and data reduction costs.

The selected parameters may be divided into groups to be measured in separate tests within an experiment; for example, (1) the access and disengagement parameters, and (2) the user information transfer parameters. The separate groups of parameters should be measured under comparable conditions, in the same experiment, if they are to be stated together in a measurement report.

2.3 Population Definition. The third step in designing a performance measurement experiment is to define the population of performance trials, such as access attempts, on which the measurements will focus (and to which the measured values will refer). This requires that the following items of information be specified:

(1) Characteristics of the user pairs to which the data communication service is provided

(2) Overall set of user pairs to be represented (as distinguished from the subset actually measured)

(3) Observation period (or periods) within which performance is to be characterized (and within which tests may be conducted)

(4) Characteristics of the user/system interfaces to be monitored in collecting the performance data, including the placement of measurement points

(5) Session profiles (or equivalent specifications) defining the event sequences that occur at the monitored interfaces during a typical (successful) data communication session of the type to be observed; and any service refusal (e.g., blocking) or service interruption (e.g., pre-emption) sequences explicitly allowed by the system design

(6) Reference events corresponding to each defined interface event, and the data communication session type

(7) Timeouts and thresholds that distinguish successful trials from performance failures

Items (1) through (3) delimit the population of interest in space and time. Items (4) and (5) specify the measurement interfaces and associated protocol events. Items (6) and (7) specialize the general parameter definitions to the specified interfaces and protocols.

Appendix B provides an example of the specification of this information in a particular (hypothetical) experiment design. The data communication session profile used in designing an early performance measurement in accordance with ANSI X3.102-1983 is illustrated in [3]. Detailed definitions for the reference events to be identified in item (6) are provided in Section 3 of this standard. Timeouts and thresholds to be used in distinguishing successful trials from performance failures shall be as defined in ANSI X3.102-1983.

To avoid bias, the population must be defined in such a way that each individual trial can be given equal consideration and weight in the estimation of population parameters. In most experiments, "equal consideration and weight" is achieved by random sampling (discussed more fully in 2.5.1 and [2]). The cost implications of random sampling often limit the population that can be considered in an experiment; i.e., the selected population is the largest set of performance trials over which random sampling is economically feasible. Note that in characterizing a multiuser network, the overall population of interest is the set of trials involving all user pairs; i.e., variability both within and between user pairs must be considered.

2.4 Performance Factors. The fourth step in designing a performance measurement experiment is to specify the factors (system and usage variables) presumed to influence performance, the relevant levels or values of each factor, and the factor combinations to be tested.

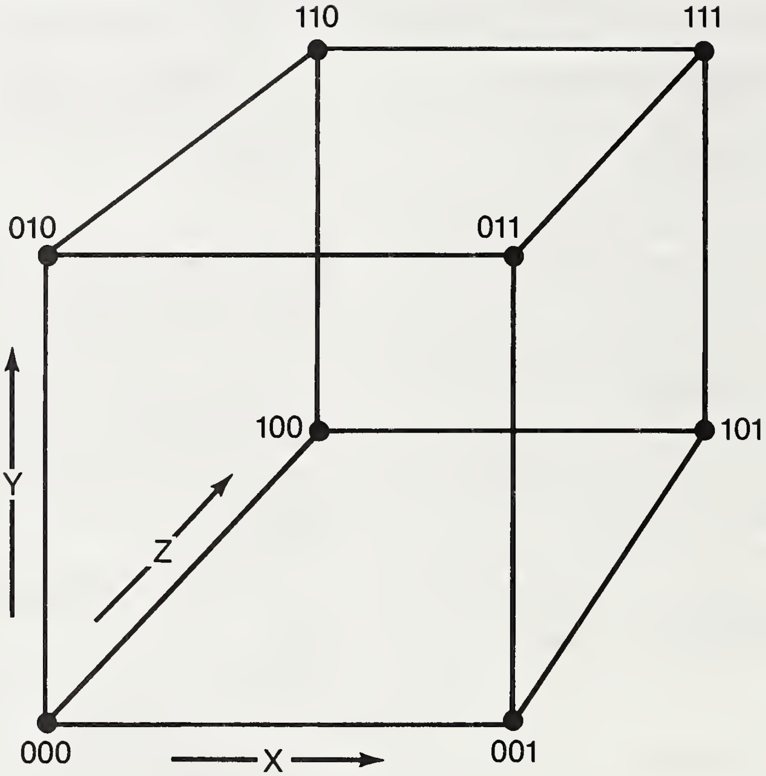
No single comprehensive list of relevant factors, levels, and factor combinations can be defined, since the appropriate choices depend on the experiment context and objectives. Table 2 lists a few factors that are often of interest in data communication performance measurements, and identifies typical levels for each. Note that the levels may be either qualitative or quantitative. A simple way of envisioning (and representing) 2-level factor combinations is with standard binary notation, as shown for three 2-level factors in Figure 4.

The selection of performance factors, levels, and factor combinations in a particular experiment should be guided by the following principles:

(1) Performance factors and levels should be distinguished in an experiment design only

Table 2
Example Performance Factors and Levels

PERFORMANCE FACTOR	TYPICAL LEVELS
NETWORK ALTERNATIVE	NETWORK A NETWORK B
LINE SPEED	9600 BITS PER SECOND 56000 BITS PER SECOND
SWITCHING TECHNOLGY	CIRCUIT SWITCHING PACKET SWITCHING
TRANSMISSION FACILITIES	SATELLITE TERRESTRIAL
ROUTING ALGORITHM	FIXED DYNAMIC
TERMINAL PROTOCOLS	OSI TRANSPORT CLASS 2 OSI TRANSPORT CLASS 4
ERROR CONTROL SCHEME	FORWARD ERROR CORRECTION AUTOMATIC REPEAT REQUEST
ACCESS PRIORITY	NORMAL EXPEDITED
USER BLOCK SIZE	64 BYTES 512 BYTES
TRAFFIC	BUSY HOUR NON-BUSY HOUR
VALUE ADDED FEATURES (E.G. CODE CONVERSION, RATE ADAPTION)	PROVIDED NOT PROVIDED



Factor Combinations

Factors		
Z	Y	X
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

Factor Levels (0 or 1)

Figure 4
Binary Representation of Factor Combinations for Three 2-Level Factors

if their effects must be specifically determined in order to achieve the experiment objectives.[3]

(2) Where feasible, each defined factor combination should be tested at least once; and the entire experiment should be replicated to test for significant unaccounted factors.

(3) In cases where the number of defined factor combinations is too large to permit the testing of each, the tested factor combinations should be chosen so as to provide maximum accuracy in comparing factor levels whose performance effects are expected to be most important. In general, the selected factor combinations should include combinations that differ only in these critical factor levels.

An experiment in which every possible combination of the defined factor levels is tested at least once is called a **full factorial** experiment. An experiment in which some of the possible factor combinations are not tested is called a **fractional factorial** experiment. As a simple illustration of the latter, consider the cube of Figure 4. The eight factor combinations represented there could be reduced to the four combinations 010, 001, 100, and 111 (at opposite corners of the cube) in a fractional factorial experiment, and the results would still provide information on each factor effect. What is lost in such an experiment is the ability to identify **interactions** among factors; i.e., joint effects that differ from the sum of the individual factor effects. Further discussion of these principles is provided in [2].

2.5 Population Sample. The fifth step in designing a performance measurement experiment is to select from the defined population a representative sample of performance trials (e.g., call attempts) to be tested. Two considerations are involved: the method of selection and the sample size.

2.5.1 Method of Selection. The basic principle to be followed in selecting performance trials to be included in a measurement sample is that of randomization; i.e., the particular trials selected to represent a population should, within practical constraints, be a random sample of the entire population. For the purposes of this standard, a **random sample** of a statistical population is a subset of the population chosen in such a way that each performance trial has an equal chance of being included in the sample. Random samples may be selected with the aid of random number tables as outlined, for example, in [2].

Practical constraints often require a departure from complete randomization in the selection of the trials to be tested in an experiment. Once a particular data extraction capability has been established, it is efficient to observe many trials before changing to a different arrangement; but the consecutive trials are not then selected from the population at random. Certain users may require continuous service availability, and thus may not be tolerant of test interruptions. A random selection may pair users that are so close geographically or topologically that the performance measured between them would be unrepresentative. In many applications, the sampled population is not homogeneous, but consists of several more or less distinct subsets (associated, for example, with different user types or time intervals) that should be sampled equally.[4] Random selection may also (with low probability) result in samples that are clearly unbalanced with regard to the measured population (e.g., repeated sampling of the same user pairs in an experiment intended to characterize a multiuser network).

Dependence among performance trials can be accounted for in the test data analysis by, in effect, reducing the number of trials counted. This process is described in [4]. Other necessary departures from randomization may be addressed in two ways. The first is to impose constraints on the random selection of performance trials before sampling. An example is a three-stage sampling plan, in which the first stage selects geographical areas, the second stage selects individual users within each area, and the final stage selects particular trials, such as calls, involving the selected users. Such a plan is described in [5]. The second is to select the sample without restriction, but reject statistically

---

[3] As noted earlier, only one factor combination is examined in absolute performance characterization and simple hypothesis test experiments.

[4] Such subsets are called "blocks" in the experiment design literature.

unbalanced samples. The former approach is recommended.

2.5.2 Sample Size. Specific procedures and formulas (and an available computer program) for test sample size determination are presented in [4]. As discussed there, test sample sizes may be determined in either of two ways:

(1) They may be derived from measurement precision objectives

(2) They may be specified on the basis of practical constraints (e.g., data storage capacity or the reasonable duration of an individual test)

In experiments involving replication or the testing of several factor combinations, the overall sample size should be increased accordingly. In experiments designed to characterize a multiuser network, sample size determination typically involves two steps:

(1) Selecting a subset of user pairs to be tested from the (possibly much larger) set of user pairs the network interconnects

(2) Choosing the number of performance trials to be observed in testing each selected user pair

A simple example of this two-step process is provided in Appendix B. Further information is provided in [2].

In experiments where several parameter values are to be estimated from a common sample, the (single) sample size should be chosen so as to achieve the most stringent precision requirement; i.e., the precision requirement demanding the largest sample size. The other precision requirements will then also be achieved. Sample size requirements for parameters to be estimated from a common sample may be determined by successive entries to the computer program described in [4]. As illustrated in Appendix B, the lowest failure probability among the parameter values to be estimated frequently determines the overall sample size.

Irrespective of how the sample size is determined, a desired confidence level or significance level for the experiment results should normally be specified in the experiment design. A **confidence level** is a numerical value, typically expressed as a percentage, that describes the likelihood that a confidence interval calculated from sample data will contain the true value of the estimated parameter. The corresponding specification in hypothesis testing is the **significance level,** which can be expressed as the complement of the confidence level.

The choice of a numerical confidence (or significance) level for a performance measurement is dependent on the experiment objectives. Confidence levels of 90% and 95% (corresponding to significance levels of 10% and 5%) are commonly used. These specifications are more fully discussed in [4].

In general, the precision sought in an experiment should be determined by two factors: the cost of conducting the experiment and the potential economic impact of the resulting data.

2.6 Test Conditions. The sixth step in designing a performance measurement experiment is to specify a particular factor combination to be used in each test. As a simple example, Steps (4) and (5) of the experiment design procedure might identify two 2-level performance factors (e.g., common carrier service and access line speed) to be examined for performance effects in a total of 12 tests. The task at Step (6) would then be to define the particular service and access line speed to be used in each of the 12 tests.

Even in a relatively small experiment, the number of possible assignments of factor combinations to tests is quite large. As an example, 4 factor combinations can be assigned to 12 tests in over 14 million ways with each factor combination tested at least once. The selected experiment design represents a particular one of these many possible assignments.

Three general objectives of individual performance measurements were identified in the introduction to this section: measurement accuracy, clearly defined applicability of the measurement results, and efficient use of measurement resources. In a typical experiment design, the range of application of the measurement results is determined in Steps (2) through (5). The objective in defining the test conditions at Step (6) is then to achieve a favorable combination of measurement accuracy and efficiency.

Accuracy imposes two requirements on parameter estimates: absence of bias (no systematic measurement error) and good precision (small variability among independent measurements). As discussed in 2.5, the technique of randomization eliminates bias and also makes it possible to calculate the precision of the parameter estimates. In general, then, *factor combinations should be assigned to individual tests as randomly as possible under the constraints of a particular experiment.*

As discussed in 2.5, measurement efficiency requires restrictions on randomization in many situations. Such restrictions may arise from practical constraints on the conduct of an experiment, or from a desire to improve the precision of parameter estimates by balancing the assignment of factor combinations among distinct subsets of the population.

The most common practical constraint on randomization in the assignment of factor combinations to tests is the additional time and cost associated with setting up a given factor combination repeatedly, rather than once, for all tests of a given type. Examples of performance factors that may be difficult to vary frequently are the city of call origination, the selected common carrier service, and the access line speed. Commonly used designs that restrict randomization to accommodate such constraints are randomized blocks, balanced incomplete blocks, and Latin squares. These designs are described in [2].

2.7 Mathematical Model. It is often useful (though not essential) to describe the design of a performance measurement experiment with an explicit mathematical model. Such models provide a concise synopsis of the experiment design and a basis for estimating measurement precision and performance variability in the data analysis. The simple mathematical models addressed in this standard relate four statistical quantities:

(1) An individual observed value of the performance variable in question (e.g., block transfer time)

(2) The true (but unknown) population value of the corresponding performance parameter

(3) Factor effects observed in the experiment

(4) Random errors

Individual observed values of the performance variable are expressed as a function of the other three quantities. In the case of absolute performance characterization and simple hypothesis test experiments, factor effects are not considered and this function may take the following simplified form:

$$Y_i = \mu + \epsilon_i,$$

where

Y_i = The value measured in the i-th observation

μ = The parameter's true (population) mean

ϵ_i = The experimental error in the i-th observation

Such a model might be used, for example, in describing a measurement of block transfer time in an absolute performance characterization experiment.[5]

Performance factors may or may not be quantifiable, as illustrated in Table 2. Experiments involving several levels of a single, nonquantifiable factor may be modeled by an equation of the form:

$$Y_{ij} = \mu + a_j + \epsilon_{ij},$$

where

Y_{ij} = The value measured in the i-th observation at factor level j

μ = The parameter's true (population) mean

a_j = The performance effect of a particular factor level j

ϵ_{ij} = The experimental error in the i-th observation at level j

If the factor levels are quantifiable, the factor effects can be described with a regression model of the form:

$$Y_i = a + bx_i + \epsilon_i$$

---

[5]The full description of a model includes particulars about its components such as, for example, that the ϵ_i have mean zero, variance σ^2, and normal distribution, and are independent.

where

Y_i = The value of the dependent variable measured in the i-th observation

a = A constant (the intercept of the regression line)

b = A constant (the slope of the regression line)

x_i = The value of the (quantifiable) factor level in the i-th observation

ϵ_i = The experimental error in the i-th observation

The use of mathematical models in describing performance measurements is recommended. Further information on the use of mathematical models in experiment design is provided in [2].

3. Data Extraction

This section specifies functional requirements for data extraction systems designed to obtain the raw performance data needed to calculate estimated values for the parameters defined in ANSI X3.102-1983. The requirements are specified from the point of view of a single, generic "interface monitor" -- the data extraction system element associated with a particular monitored interface.

As noted in Section 1, the measurement requirements specified in this standard address the most stringent type of application, in which all 21 parameters are to be measured via continuous, independent observation of events at two or more distant interfaces during a performance measurement period. Such applications require, at each interface, an independent interface monitor that performs three major functions:

(1) *Interface Event Collection* - the detection and interpretation of transferred signals and the time-stamping of associated interface events

(2) *Event Processing* - the mapping of system-specific interface events into the system-independent reference events described in ANSI X3.102-1983

(3) *Reference Event Recording* - the creation of a performance data file (or files) recording (a) the nature and time of occurrence of each reference event; and (b) the content of each transferred user information block (where required)

The following subsections specify the detailed requirements associated with each of these functions. An existing data extraction software system designed in accordance with these requirements is described in [3].

It is also noted in Section 1 that particular applications may be much simpler. For example, there may be only one interface monitor, which observes events at each of two collocated interfaces; only certain events, needed to estimate parameters of interest, may be collected and processed; and the data reduction phase of a measurement may be performed on-line, eliminating the need to record individual reference events. Users of this standard should interpret and properly restrict the requirements specified in this section in accordance with their particular application.

3.1 Interface Event Collection. The data extraction phase of the performance measurement process begins with the collection of interface events at physical or functional boundaries between communicating users and a monitored data communication system or subsystem. During a particular performance measurement period, an interface monitor shall (1) detect all signals transferred across its associated interface (in either direction) during a specified performance measurement period; (2) interpret the transferred signals as a sequence of discrete events, each having a specific meaning within the monitored interface protocol; and (3) determine the time of occurrence of each such interface event. This subsection develops these requirements by describing the various types of interfaces that may be monitored and by defining the interface event concept. Two general types of measurement interfaces are defined: the user/system interface and the subsystem interface. Examples of discrete events occurring at each interface are provided.

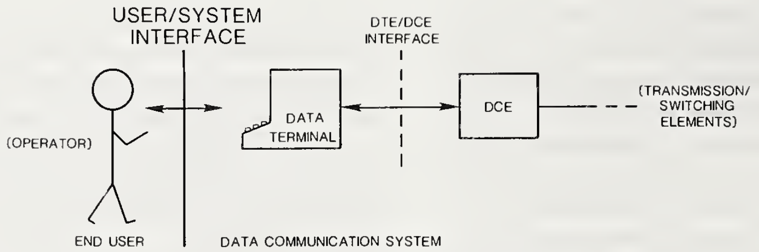
3.1.1 User/System Interfaces. In ANSI X3.102-1983, the **end user** of a data communication system or service is defined as either (1) the human operator of a data terminal, with any associated data medium; or (2) a computer application program that utilizes communicated information, with any associated data medium. An example of the first type of end user is a person operating an automated banking terminal. Examples of the second type of end user are (1) a FORTRAN program that calculates payroll information based on employee records stored in a remote data base; (2) the remote data base management program that provides the employee records; and (3) an industrial process control program that regulates the operation of an attached machine under the direction of a remote plant control system. Typical data media associated with terminal operators are magnetic disks, magnetic stripe cards, and typewritten or printed pages. Typical data media associated with application programs are magnetic tape and magnetic disks. Such media are not necessarily collocated with the user, and may also include physical phenomena that interact with remote sensors or effectors.

In ANSI X3.102-1983, a **data communication system** (more briefly, a **system**) is defined as a collection of transmission facilities and associated switches, data terminals, and protocols that provide data communication service between two or more end users. A data communication system includes all functional and physical elements that participate in transferring information between end users. The particular system element that interfaces with the end user depends on the type of user. The system element that interfaces with a human operator or associated data medium is a data terminal equipment (DTE). In most cases, the system element that interfaces with a computer application program is the computer's operating system. An application program may also communicate directly with a data communication hardware or software element (e.g., in computers that do not have an operating system).

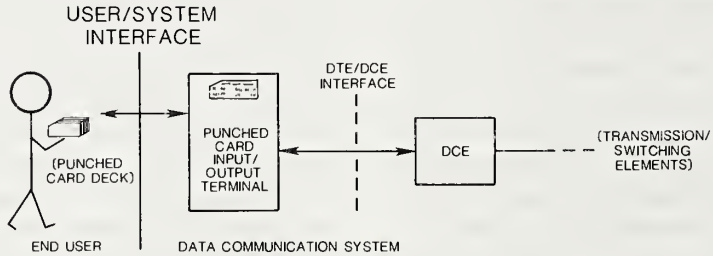
Any physical or functional boundary between an end user and a data communication system is called a **user/system interface**. Figure 5 illustrates the four major types of user/system interfaces. When the end user is a human operator with no associated data medium, the user/system interface is defined to be the physical interface between the operator and the data terminal (Figure 5(a)). When the end user is a human operator with an associated data medium, the user/system interface is defined to include both the physical interface between the operator and the data terminal, and the physical interface between the medium and its input/output terminal (Figure 5(b)). When the end user is an application program with no associated (separate) data medium, the user/system interface is defined as the functional interface between that program and the local operating system or communication access program (Figure 5(c)). When the end user is an application program with an associated (separate) data medium, the user/system interface is defined to include both the previously described functional interface and the physical interface between the medium and its input/output terminal (Figure 5(d)). The user/system interface is also sometimes called the **end user interface**.

3.1.2 Subsystem Interfaces. This standard is primarily intended to be used in measuring performance between end user interfaces. It may also be used to measure the performance of a group of system elements, terminated at digital interfaces, comprising a portion of a data communication system. Any such group of elements is called a **data communication subsystem** (or simply a **subsystem**). The physical or functional boundaries delimiting a subsystem are called the **subsystem interfaces**. Each subsystem interface also identifies a collection of entities outside the subsystem, comprising one or more end users and the data communication system elements that connect those users with the subsystem. Any such collection of entities can be regarded as an **aggregate user** of the subsystem, and can be treated as a single entity in conducting subsystem performance measurements.

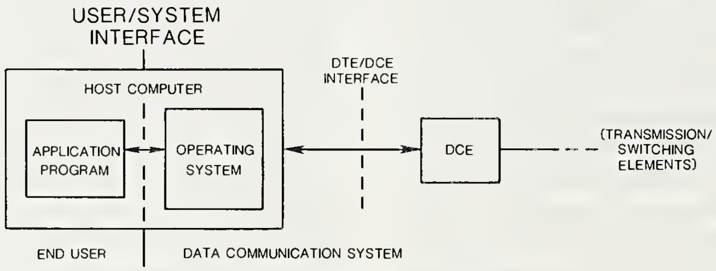
Figure 6 illustrates three typical subsystem interfaces and the associated aggregate users. In the first example (Figure 6(a)), each subsystem interface is a physical data terminal equipment/data circuit-terminating equipment (DTE/DCE) interface on a customer's premises. The data communication subsystem comprises the DCEs and the network interconnecting them (e.g., the public switched telephone network). The two aggregate users here are the DTE and operator, on one end, and the host computer, on the other.



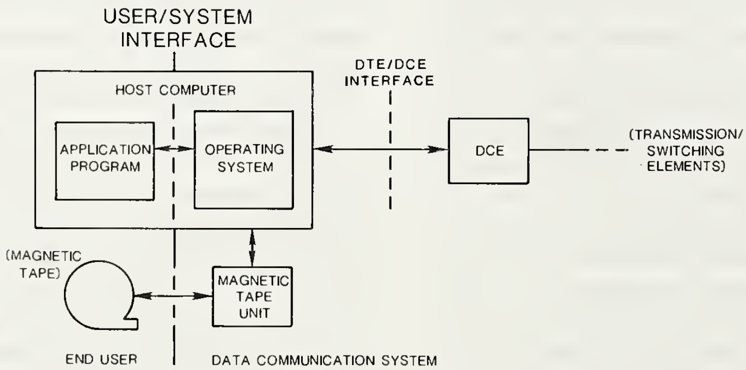
(a) Basic Operator Interface



(b) Operator Interface (with Associated Data Medium)

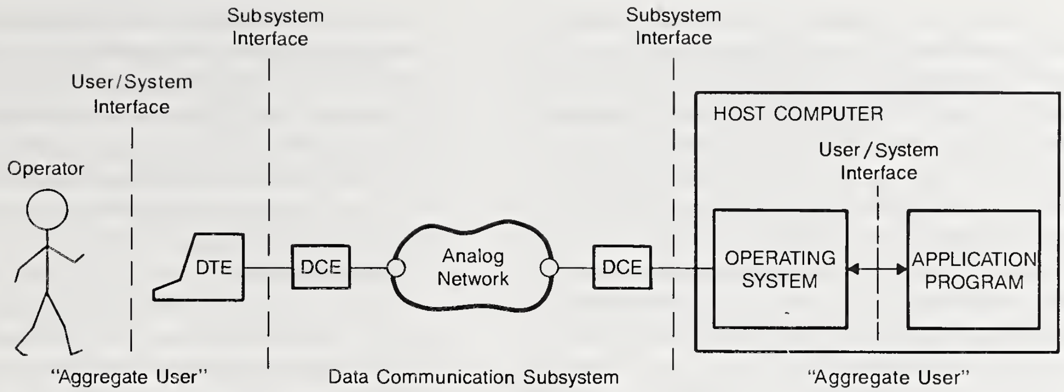


(c) Basic Application Program Interface

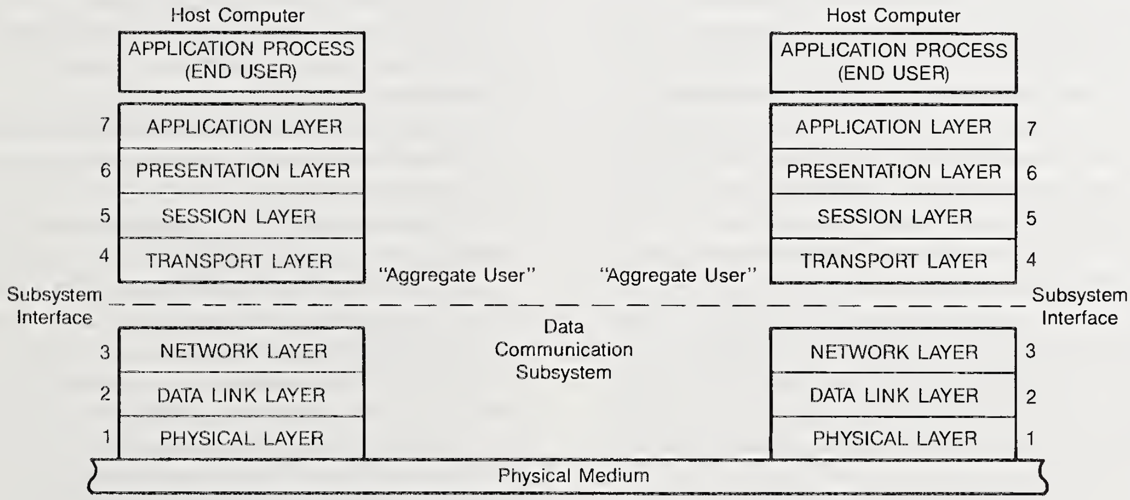


(d) Application Program Interface (with Associated Data Medium)

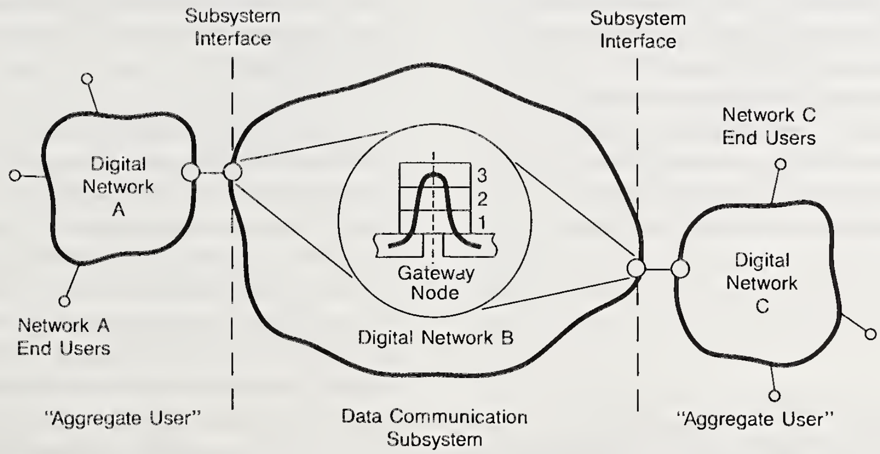
Figure 5
Interfaces between the End User and the Data Communication System



(a) DTE/DCE Interfaces Delimiting a Subsystem



(b) Protocol Layer Boundaries Delimiting a Subsystem



(c) Internetwork Gateway Interfaces Delimiting a Subsystem

Figure 6
Typical Subsystem Interfaces

In the second example (Figure 6(b)), each subsystem interface is a functional interface between adjacent communication protocols in a host computer implementing a layered protocol architecture. The data communication subsystem here comprises the network layer and lower layers and the physical communication medium interconnecting them. In each computer, the upper four protocol layers and the application process (end user) collectively are an aggregate user of the "network service" provided by the subsystem.[6]

In the third example (Figure 6(c)), each subsystem interface is within a digital transit network gateway node. It is assumed that each gateway node implements a three-layer protocol architecture, and that the intranetwork and internetwork protocols differ. The specific subsystem interfaces might then be the functional interfaces between the intranetwork and internetwork Layer 3 protocols. The data communication subsystem comprises the gateway node intranetwork protocols and all network B elements that interconnect them. The aggregate users here comprise the internetwork gateway node protocols, the physical internetwork transmission facilities, the end networks (A and C), and their end users. As in the case of the end user interfaces, the subsystem interfaces need not be identical.

3.1.3 Interface Events. Any discrete transfer of information across a user/system (or subsystem)[7] interface is called an **interface event.** Such events can occur in a variety of ways. Typical events at an operator/terminal interface are manual keystrokes and the printing or displaying of received characters. Typical events at an application program/ operating system interface are the issuance of operating system calls and the setting and clearing of flags (see Appendix B). Typical events at a medium/terminal interface are the reading and writing of magnetic disks. Typical events at a physical subsystem interface are the issuance of DTE/DCE interchange signals, such as the RS-232C Request to Send and Clear to Send signals. Typical events at a functional subsystem interface are the issuance of interlayer "service primitives" or intermodule messages.

For the purpose of defining the time of occurrence of interface events, information is defined to have been transferred from a user to the system when two conditions have been met:

(1) The information is physically present within the receiving (system) facilities

(2) The system has been authorized to send or (in the case of overhead information) process that information

Similarly, information is defined to have been transferred from the system to a user when two conditions have been met:

(1) The information is physically present within the receiving (user) facilities

(2) The user has been notified that the information is available for use

When the user/system interface is within a computer, information transfers can occur either by physical movement of the information or by transfer of right of access to the information. Several practical methods of synchronizing event time clocks in geographically remote equipment are described in [6]. The application of one such method in a performance measurement experiment conducted in accordance with ANSI X3.102-1983 is described in [3].

3.2 Event Processing. The interface events discussed in 3.1.3 cannot be used directly in defining user-oriented performance parameters because they are system specific; i.e., they vary from one data communication system to another. This problem is solved in ANSI X3.102-1983 by defining the standard parameters not in terms of particular system-specific interface events, but in terms of more general, system-independent **reference events.** Each reference event is a "generic event" that subsumes many system-specific interface events having a common performance significance; and each is defined in such a way that it can always be identified, if it occurs, in any particular data communication session.

---

[6]Depending on implementation, the interface events at a layer boundary may or may not correspond to explicit "service primitives," operating system calls, or intermodule messages. Performance measurement at protocol layer boundaries may be impractical if no such correspondence exists.

[7]For brevity, "subsystem" is not explicitly stated as an alternative to "system" in most subsequent references in this standard.

The reference events collectively specify all information needed to describe performance in a comprehensive, user-oriented way.

An example will help to clarify the relationship between system-specific interface events and the associated reference events. An end user's action in lifting a telephone handset off-hook transfers one bit of information (the new hookswitch position) from the user to the system. That transfer is a system-specific interface event that notifies the system of a user's need for service in the public switched telephone network. Completely different interface events may convey that information in other networks (e.g., typing a Connect request at a data terminal to access a packet switching network).

All interface events that communicate a user's need for data communication service or otherwise initiate a data communication session are represented by the single reference event, Access Request. This reference event is used to define the beginning of the access function and to start the counting of access time. Defining the access parameters in terms of this and similar reference events makes the parameters system independent, and makes their values comparable between services with different user interface protocols.

Successful measurement of performance parameters depends on a proper translation of the system-specific events observed at a monitored user/system interface into corresponding system-independent reference events. The second major function of the generic interface monitor described here is to perform that translation. That event-processing requirement is developed in this subsection by defining the reference events and giving examples of system-specific counterparts to each. A total of 15 reference events (9 "primary" reference events and 6 "ancillary" reference events) are discussed.

3.2.1 Primary Reference Events. The "primary" reference events are used in defining the primary parameters described in ANSI X3.102-1983. Table 3 lists the nine primary events and identifies, for each event, a system-specific counterpart that might be observed in monitoring the end user interfaces during communications between a terminal operator and a remote application program via a packet switching network (e.g., the original ARPANET).[8] The nine events are associated with the three primary communication functions identified earlier: access, user information transfer, and disengagement. These events are fully defined in ANSI X3.102-1983.

The **Access Request** notifies the system of a user's desire to initiate a data communication session. It begins the access function and starts the counting of access time. Two specific examples of Access Request events have been cited earlier -- the off-hook event in the public switched telephone network, and the typing of a Connect request by a terminal operator in the ARPANET.

The **nonoriginating** user in a data communication session is the user that does not initiate the access request. **Nonoriginating User Commitment** expresses a nonoriginating user's willingness and ability to participate in a specific requested data communication session (or, in general, any session that may be requested in the future). The occurrence of that event during (or prior to) access in a connection-oriented session eliminates Incorrect Access as a possible outcome. The most familiar example is the called user's answering of a normal telephone call. Issuance of an OPEN ANY HOST (LISTEN) system call by an application program in the ARPANET is another. The latter event occurs prior to issuance of a corresponding Access Request in most cases.

The **System Blocking Signal** is a system's way of telling a user that it cannot provide communication service on a particular request because some required system facility is unavailable. The required facility (e.g., trunk circuit) may be unavailable because it is serving another user, or because it is in an outage condition; the two possibilities often cannot be distinguished at the end user interface. The occurrence of a System Blocking Signal (within the maximum access performance time) identifies the outcome of an access attempt as Access Denial. Examples of

---

[8] The original ARPANET is used as an example here because it is static, well documented, and generally representative of modern data communication systems. Table 3 is a refinement of Table C1 in Appendix C of ANSI X3.102-1983. Transfer sample input/output events are identified as particular instances of block input/output events (events 6 and 7).

Table 3
Primary Reference Events

FUNCTION	REFERENCE EVENT	SYSTEM IMPACT	PERFORMANCE SIGNIFICANCE	ARPANET EXAMPLES
ACCESS	1. ACCESS REQUEST	REQUESTS INITIATION OF DATA COMMUNICATION SESSION AND COMMITS THE ORIGINATING USER TO PARTICIPATE.	BEGINS ACCESS FUNCTION. STARTS THE COUNTING OF ACCESS TIME.	OPERATOR TYPING OF CONNECT REQUEST.
	2. NONORIGINATING USER COMMITMENT	IN A CONNECTION-ORIENTED DATA COMMUNICATION SESSION, INDICATES NONORIGINATING (CALLED) USER WILLINGNESS TO PARTICIPATE.	ELIMINATES INCORRECT ACCESS AS A POSSIBLE ACCESS OUTCOME.	APPLICATION PROGRAM ISSUANCE OF OPEN ANY HOST (LISTEN) SYSTEM CALL.
	3. SYSTEM BLOCKING SIGNAL	NOTIFIES ORIGINATING USER THAT THE SYSTEM CANNOT SUPPORT A REQUESTED DATA COMMUNICATION SESSION.	IDENTIFIES ACCESS ATTEMPT OUTCOME AS ACCESS DENIAL.	SYSTEM PRINTING OF NET TROUBLE MESSAGE AT OPERATOR TERMINAL.
	4. USER BLOCKING SIGNAL	NOTIFIES SYSTEM THAT THE ISSUING USER WILL NOT SUPPORT A REQUESTED DATA COMMUNICATION SESSION.	IDENTIFIES ACCESS ATTEMPT OUTCOME AS USER BLOCKING (EXCLUDED FROM SYSTEM PERFORMANCE MEASUREMENT.)	OPERATOR TYPING OF CLOSE REQUEST (DURING ACCESS).
	5. START OF BLOCK INPUT TO SYSTEM	TRANSFERS ONE OR MORE BITS AT BEGINNING OF USER INFORMATION BLOCK FROM SOURCE USER TO SYSTEM.	WHEN BLOCK IS THE FIRST BLOCK IN A DATA COMMUNICATION SESSION (AFTER NONORIGINATING USER COMMITMENT IN CONNECTION-ORIENTED SESSIONS), COMPLETES ACCESS FUNCTION AND BEGINS USER INFORMATION TRANSFER. STOPS THE COUNTING OF ACCESS TIME	OPERATOR TYPING OF FIRST USER INFORMATION CHARACTER AT A BUFFERED CRT TERMINAL.
USER INFORMATION TRANSFER	6. START OF BLOCK TRANSFER	AUTHORIZES THE SYSTEM TO TRANSMIT A GIVEN USER INFORMATION BLOCK.	(1) BEGINS BLOCK TRANSFER FUNCTION AND STARTS THE COUNTING OF BLOCK TRANSFER TIME. (2) WHEN BLOCK PRECEDES THE FIRST BLOCK IN A TRANSFER SAMPLE, BEGINS COLLECTION OF THE SAMPLE AND STARTS THE COUNTING OF SAMPLE INPUT TIME. (3) WHEN BLOCK IS THE LAST BLOCK IN A TRANSFER SAMPLE, COMPLETES INPUT OF SAMPLE AND STOPS THE COUNTING OF SAMPLE INPUT TIME.	OPERATOR TYPING OF ANY USER INFORMATION CHARACTER AT AN UNBUFFERED SOURCE TERMINAL. TYPING OF CARRIAGE RETURN AT A BUFFERED SOUCE TERMINAL.
	7. END OF BLOCK TRANSFER	TRANSFERS A GIVEN USER INFORMATION BLOCK TO THE DESTINATION USER, WITH APPROPRIATE NOTIFICATION TO THAT USER WHERE REQUIRED.	(1) COMPLETES BLOCK TRANSFER FUNCTION AND STOPS THE COUTING OF BLOCK TRANSFER TIME. (2) WHEN BLOCK PRECEDES THE FIRST BLOCK IN A TRANSFER SAMPLE, BEGINS OUTPUT OF THE SAMPLE AND STARTS THE COUNTING OF SAMPLE OUTPUT TIME. (3) WHEN BLOCK IS THE LAST BLOCK IN A TRANSFER SAMPLE, COMPLETES COLLECTION OF THE SAMPLE AND STOPS THE COUNTING OF SAMPLE OUTPUT TIME.	SYSTEM PRINTING OR DISPLAY OF COMPETE SOUCE USER INFORMATION BLOCK AT THE DESTINATION TERMINAL.
DISENGAGEMENT	8. DISENGAGEMENT REQUEST	REQUESTS TERMINATION OF A USER'S PARTICIPATION IN A DATA COMMUNICATION SESSION.	BEGINS DISENGAGEMENT FUNCTION. STARTS THE COUNTING OF DISENGAGEMENT TIME.	OPERATOR TYPING OF CLOSE REQUEST (AFTER SUCCESSFUL ACCESS).
	9. DISENGAGEMENT CONFIRMATION	CONFIRMS TERMINATION OF A USER'S PARTICIPATION IN A DATA COMMUNICATION SESSION	COMPLETES DISENGAGEMENT FUNCTION. STOPS THE COUNTING OF DISENGAGEMENT TIME.	SYSTEM PRINTING OF CLOSE MESSAGE AT OPERATOR TERMINAL.

NOTE. Interface events may have no corresponding primary reference event. Such events may be represented by the primary event number "0" in recording reference event sequences.

System Blocking Signals are issuance of the "circuit busy" signal to a calling user in the public switched telephone network, and the system's printing of a NET TROUBLE message at a calling operator's terminal in the ARPANET.

The **User Blocking Signal** is the user's counterpart to a System Blocking Signal. Its issuance by a user during access indicates that the issuer will not participate in a requested data communication session, and the access attempt should therefore be abandoned. The occurrence of a User Blocking Signal (within the maximum access performance time) identifies the outcome of an access attempt as User Blocking; this outcome is excluded from system performance measurement. Examples of User Blocking Signals are a calling user's replacing the handset on hook (during connection establishment) in the public switched telephone network, and the called user's issuance of a Close request during a call establishment attempt in the ARPANET.

The **Start of Block Input to System** transfers one or more bits at the beginning of a user information block from the source user to the data communication system. That event usually coincides with the start of transfer of the block (i.e., the occurrence of the event defined next). However, the two events differ in cases where the system provides an input buffer (e.g., the one-line input buffer provided in many CRT terminals). In such cases, start of input is defined to occur when the first bit of user information is physically stored in that buffer. An example is the operator's typing of the first user information character at a buffered CRT terminal. When the input block is the first block in a data communication session (after nonoriginating user commitment in connection-oriented sessions), Start of Block Input completes the access function, stops the counting of access time, and identifies the start of user information transfer. The reason for using the start of input of user information (rather than a "connection open" or similar system response) to define the end of access is that such responses are not provided in all systems.

The **Start of Block Transfer** initiates actual transmission of a specified user information block between the source and destination users. That event occurs, for any given user information block,[9] when (1) all bits in the block are physically present within the system facility, and (2) the system has been authorized to transmit that information. Authorization may either be an explicit user action (e.g., typing Carriage Return at a buffered CRT terminal) or an implicit part of inputting the user information itself (e.g., typing a single character at an unbuffered asynchronous terminal). For each block, the Start of Block Transfer event begins the counting of block transfer time. When the transmitted block precedes the first block in a transfer sample, Start of Block Transfer begins the collection of that sample and starts the counting of sample input time (a transfer sample includes the interblock gap preceding the first block in the sample). When the transmitted block is the last block in a transfer sample, Start of Block Transfer completes input of the sample and stops the counting of sample input time.

The **End of Block Transfer** completes the transmission of a specified user information block between the source and destination users. That event occurs, for any given user information block, when (1) all bits in the block are physically present within the destination user facility, and (2) the destination user has been notified that the information is available for use. The notification may be explicit or implicit. For each block, the End of Block Transfer event stops the counting of block transfer time. When the received block precedes the first block in a transfer sample, End of Block Transfer begins output of that sample and starts the counting of sample output time. When the received block is the last block in a transfer sample, End of Block Transfer completes the collection of the sample and stops the counting of sample output time. An example of an End of Block Transfer event is the completion of printing of a line of text at a destination terminal.

The **Disengagement Request** notifies the system of a user's desire to terminate an established data communication session. It is

[9]As defined in ANSI X3.102-1983, a **user information block** is a contiguous group of bits delimited at a source user/system interface for transfer to a destination user as a unit. The user information block may be defined to correspond to a time interval (e.g., second or decisecond) in cases where the transmission rate is fixed.

complementary to the access request in most systems. Each Disengagement Request begins the disengagement function and starts the counting of disengagement time for one or both users. Disengagement is normally requested simultaneously for both users in connection-oriented sessions, and independently for each end user in connectionless sessions. Examples of Disengagement Requests are a user's hanging up in the public switched telephone network, and the terminal operator's typing of a Close request (after successful access) in the ARPANET.

Disengagement Confirmation verifies that a particular user's participation in an established data communication session has been terminated. For that user, it completes the disengagement function and stops the counting of disengagement time. Disengagement Confirmation occurs, for a particular user, when (1) disengagement of that user has been requested; and (2) the user is able to initiate a new access attempt. Most data communication systems notify the user that a new access attempt can be initiated by issuing an explicit Disengagement Confirmation signal. An example is the system's printing of a Closed message at an operator terminal in the ARPANET. In cases where no Disengagement Confirmation signal is issued, the user may initiate a new access attempt to confirm disengagement.

As noted earlier, the parameters defined in ANSI X3.102-1983 may be used to characterize data communication performance at subsystem interfaces, such as the DTE/DCE interfaces. Table 4 associates each of the nine primary reference events defined in this section with specific events observed at the DTE/DCE interfaces in the X.25 (Virtual Call) and X.21 protocols. The X.25 interface events are assumed to be observed at the Layer 3 input/output queues.

Most, but not all, observed interface events will have performance significance in accordance with ANSI X3.102-1983. Call Progress signals often do not. In some measurement situations, a single interface event translates into two reference events. The second event is most often an ancillary event, as discussed in 3.2.2.

3.2.2 Ancillary Reference Events. The second group of measurement events to be defined are the "ancillary" reference events. These events are used in defining the ancillary parameters, which describe the influence of user delays on the primary "speed" parameter values. A record of these events is also needed to identify the entity responsible for performance "timeout" failures [7].

The ancillary reference events classify (and represent) observed interface events with respect to their effect on user and system "responsibility" for the generation of future events. The occurrence of an event at a particular interface may affect the responsibility state at that (local) interface, at the remote interface, or both. An event may have any one of three local responsibility effects:

(1) The event may leave the system responsible for generating the next event at the local interface.

(2) The event may leave the local user responsible for generating the next event at the local interface.

(3) The event may (temporarily) relieve both the user and the system of responsibility for generating a subsequent event at the local interface (because the next event in the normal event sequence occurs at the remote interface).

A calling user's off-hook action illustrates the first effect. That event makes the system responsible for generating the next event (dial tone) at the calling interface. Similarly, a system's issuance of dial tone illustrates the second effect. The third effect occurs (for example) on issuing a Call Request packet in X.25. That event temporarily relieves both the calling user and the system of responsibility for generating a subsequent event at the calling interface, because the next event in the normal event sequence (delivery of the Incoming Call packet) occurs at the remote interface.

Independent of its responsibility effect at the local interface, an event may or may not give the system responsibility for generating a subsequent event at the remote interface. For example, a user's input of a data packet to the system creates a system responsibility for delivering it to the destination; whereas issuance of an X.25 Restart Request packet has no such remote effect. Combining the possible local and remote responsibility effects gives a total of six overall effects an interface event may have. These are represented by the six ancillary events listed in Table 5.

Table 4
Correspondence between System-Independent and System-Specific Events

SYSTEM-INDEPENDENT REFERENCE EVENT	X.25 INTERFACE EVENT	X.21 INTERFACE EVENT
ACCESS REQUEST	Call Request Packet Placed in Calling DTE Output Queue	Calling DTE Issuance of Call Request Signal (t = 0, c = On)
NONORIGINATING USER COMMITMENT	Call Accepted Packet Placed in Called DTE Output Queue	Called DTE Issuance of Call Accepted Signal (t=1, c = On)
SYSTEM BLOCKING SIGNAL	Clear Indication Packet Placed in Calling DTE Input Queue During Call Set-up	DCE Issuance of Call Progress Signal (with Clearing) to Calling DTE
USER BLOCKING SIGNAL	Clear Request Packet Placed in Calling or Called DTE Output Queue During Call Set-up	DTE Issuance of Clear Request Signal (t =o, c = Off) During Call Set-up
START OF BLOCK INPUT TO SYSTEM	First Bit of Data Packet Placed in Transmitting DTE Output Queue	First Bit of Transmission Block* Transferred from Transmitting DTE to DCE
START OF BLOCK TRANSFER	Last Bit of Data Packet Placed in Transmitting DTE Output Queue	Last Bit of Transmission Block* Transferred from Transmitting DTE to DCE
END OF BLOCK TRANSFER	Last Bit of Data Packet Placed in Receiving DTE Input Queue	Last Bit of Transmission Block* Transferred from DCE to Receiving DTE
DISENGAGEMENT REQUEST	Clear Request Packet Placed in Clearing DTE Output Queue During Data Transfer State	DTE Issuance of Clear Request Signal (t = o, c = Off) During Data Transfer State
DISENGAGEMENT CONFIRMATION	Clear Confirmation Packet Placed in Clearing DTE Input Queue. Clear Confirmation Packet Placed in Cleared DTE Output Queue.	Clearing (and Cleared) DTE Returned to Ready State (t =1, c = Off) After Clear Confirmation

*Assumes (e.g.) ISO 1745 Transmission Block Format

Table 5
Ancillary Reference Events

ANCILLARY EVENT	LOCAL EFFECT	REMOTE EFFECT
1	SYSTEM RESPONSIBLE	NO EFFECT
2	USER RESPONSIBLE	NO EFFECT
3	RESPONSIBILITY UNDEFINED	NO EFFECT
4	SYSTEM RESPONSIBLE	SYSTEM RESPONSIBLE
5	USER RESPONSIBLE	SYSTEM RESPONSIBLE
6	RESPONSIBILITY UNDEFINED	SYSTEM RESPONSIBLE

The interface monitor's processing of a performance significant interface event may produce a primary reference event only, an ancillary reference event only, or both. Delivery of user information in the absence of flow control illustrates the first case: The interface event marks the End of Block Transfer, but does not change responsibility at the destination user interface. Issuance of dial tone in the public switched telephone network illustrates the second case: The interface event transfers responsibility at the calling interface from the system to the user, but has no primary event significance. A calling user's issuance of an X.25 Call Request packet illustrates the third case: The interface event is an Access Request and also relieves both the user and the system of responsibility for generating a subsequent event at the local interface (ancillary event 6).

Ancillary events may be implicit in some systems. As an example, a user entering characters at an asynchronous teletypewriter needs to separate successive keystrokes by a delay at least as long as the character modulation time; i.e., the reciprocal of the terminal "character rate" R_c. The system (terminal) is thus "responsible" for the first $1/R_c$ seconds after typing of each character, while the user is responsible thereafter. Expiration of the $1/R_c$ delay is an implicit ancillary event (ancillary event 2), since it is not associated with any actual interface signal.

3.3 Reference Event Recording. To measure the complete set of performance parameters described in ANSI X3.102-1983, three basic elements of performance information should be recorded (or predetermined):

(1) A set of reference events defining the performance significance of each system-specific interface signal observed at the monitored user/system interface during the performance measurement period. Both primary and ancillary events should be recorded. Where a primary and an ancillary event coincide, one event record (defining both events) may be produced.

(2) The time of occurrence (absolute or relative) of each reference event.

(3) Sufficient information about the content of the transmitted and received data to enable the detection of user information error, loss, or addition. When comprehensive measurements are required during operational service usage, the exact binary content (or binary representation) of each transmitted and received user information block should be recorded for comparison purposes.

The user information recorded at the source user and destination user interfaces may differ in situations where code conversion is performed within the system. Further, the user information may appear in the form of nonbinary symbols when measurements are conducted at the end user interfaces. To accommodate such situations, separate definitions for source user and destination user information bits are provided in ANSI X3.102-1983. A summary of these definitions follows.

(1) Source user information bits are those bits used for the binary representation of user information transferred from a source user to a data communication system for ultimate delivery to a destination user. When the user information is input as nonbinary symbols (e.g., keyboard entries) the source user information bits are the bits used to encode these symbols initially. Any bits added to this initial encoding for purposes of error control, flow control, polling, and the like are overhead bits rather than source user information bits.

(2) Destination user information bits are those bits used for the binary representation of user information transferred from a data communication system to a destination user. When the user information is output as nonbinary symbols (e.g., alphanumeric characters), the destination user information bits are the bits on which a final decoding is performed to generate the delivered symbols.

The interface monitor shall generate appropriate binary representations for the transferred user information in cases in which the user information is input or output as nonbinary symbols, and shall map the recorded source symbols into the destination code in cases in which the source and destination codes differ.

Depending on the objectives of a particular measurement, it may or may not be desirable to record the additional data needed to detect Misdelivered Blocks. It is possible to detect Misdelivered Blocks by observations at only two user/system interfaces, but the process requires that the source interface monitor create a separate user information record containing all blocks transmitted by the source user to users other than the monitored destination user during the measurement period. These blocks are compared with any Extra Blocks received by the monitored destination user to detect misdelivery, as discussed in 4.2.

Because of the additional burden its measurement imposes, Block Misdelivery Probability is described as an optional parameter in ANSI X3.102-1983. If the procedures needed to detect Misdelivered Blocks are not implemented in a particular measurement, any Misdelivered Blocks are simply counted as Extra Blocks.

4. Data Reduction

This section specifies functional requirements for data reduction systems designed to transform the performance data output by an associated data extraction system into a set of estimates for the parameters defined in ANSI X3.102-1983. The section is comprised of four parts. The first three specify procedures for processing recorded primary reference events (and associated user information records) to estimate values for the access, user information transfer, and disengagement parameters, respectively. The fourth specifies a procedure for processing recorded ancillary reference events to determine, for an associated primary function, the performance time that is attributable to user delays.

As noted in Section 1, the measurement requirements specified in this standard address the most stringent type of application, in which all 21 parameters defined in ANSI X3.102-1983 are measured via continuous, independent observation of events at two or more distant interfaces during a performance measurement period. Accordingly, the requirements specified in this section assume that data reduction is performed off-line, on the basis of comprehensive reference event histories and user information files recorded separately at the monitored user/system interfaces; that every defined performance outcome is distinguished; and that the user fraction of each access, user information transfer, and disengagement performance trial is determined. An available machine-independent data reduction computer program which implements these maximum requirements is described in [3].

Section 1 also points out that particular applications may be much simpler. For example, data reduction may be performed on-line, by incrementing event counters and timers; performance outcomes within certain categories (e.g., the four access failure outcomes) may be grouped or disregarded; and calculation of user fractions may be omitted. Users of this standard should interpret and properly select the requirements specified in this section in accordance with their particular application.

The data reduction procedures are specified using three descriptive techniques: functional flowcharts, outcome diagrams, and mathematical formulas. The functional flowcharts define a logical method of determining the outcomes of a monitored set (or sample) of performance trials (e.g., access attempts). The outcome diagrams and mathematical formulas ensure uniformity in translating the observed outcomes into parameter values.

The following notational conventions are used in this standard (Table 6):

(1) Each function is represented by a lowercase mnemonic symbol (e.g., "a" for access, "b1" for bit transfer, and so on). The various performance outcomes are represented by subscripted lowercase symbols (e.g., "a_s" for Successful Access, "$b1_e$" for Incorrect Bit, and so on).

(2) The total numbers of trials and outcomes observed during a performance measurement period are represented by corresponding uppercase letters (e.g., "A" and "A_s" for total access trials and total Successful Access outcomes, respectively).

(3) Probabilities, average performance times, average user performance times, and time rates are represented by the symbols P( ), W( ), U( ), and R( ), respectively. The symbol F( ) represents the average user fraction of performance time U( )/W( ). Individual event times are represented by the symbol t( ). The argument ( ) in each case is the performance outcome of interest. For example, the expression $W(a_s)$ denotes the average elapsed time to Successful Access.

(4) The lowercase symbols w( ) and u( ) are used to distinguish performance time and user

Table 6
Symbol Conventions

SYMBOL CATEGORY / PRIMARY FUNCTION/OUTCOME	SUPPORTING SYMBOLS										PARAMETER SYMBOLS						
	Function Symbol	Outcome Symbol	Total Trials	Total Outcomes	Total Trials Less User Dependent Failures	Starting Time	Successful Completion Time	Individual Trial Performance Time	Individual Trial--User Performance Time	Average User Performance Time	Outcome Probability	Average Performance Time	Average User Fraction of Performance Time	Outcome Rate	Specified Average Performance Time (or Rate)	Specified Average User Fraction of Performance Time	Transfer Denial Threshold
ACCESS	a		A		A′	t(a)											
Successful		a_s		A_s			$t(a_s)$	$w(a_s)$	$u(a_s)$	$U(a_s)$		$W(a_s)$	$F(a_s)$		$W_N(a_s)$	$F_N(a_s)$	
Incorrect		a_m		A_m							$P(e_m)$						
Denial		a_ℓ		A_ℓ							$P(a_\ell)$						
Outage		a_o		A_o							$P(a_o)$						
User Blocking		a_f		A_f													
BIT TRANSFER	b1		B1		B1′												
Successful		$b1_s$		$B1_s$										$R(b1_s)$	$R_N(b1_s)$		$R_T(b1_s)$
Incorrect		$b1_e$		$B1_e$							$P(b1_e)$						$P_T(b1_e)$
Misdelivered		$b1_m$		$B1_m$							$P(b1_m)$						
Lost		$b1_\ell$		$B1_\ell$							$P(b1_\ell)$						$P_T(b1_\ell)$
Extra		$b1_x$		$B1_x$							$P(b1_x)$						$P_T(b1_x)$
Refused		$b1_f$		$B1_f$													
BLOCK TRANSFER	b2		B2		B2′	t(b2)											
Successful		$b2_s$		$B2_s$			$t(b2_s)$	$w(b2_s)$	$u(b2_s)$	$U(b2_s)$		$W(b2_s)$	$F(b2_s)$		$W_N(b2_s)$	$F_N(b2_s)$	
Incorrect		$b2_e$		$B2_e$							$P(b2_e)$						
Misdelivered		$b2_m$		$B2_m$							$P(b2_m)$						
Lost		$b2_\ell$		$B2_\ell$							$P(b2_\ell)$						
Extra		$b2_x$		$B2_x$							$P(b2_x)$						
Refused		$b2_f$		$B2_f$													
SAMPLE* TRANSFER	b3		B3		B3′												
Successful — Input								$w(b3_i)$									
Successful — Output								$w(b3_o)$									
Successful — Input/Output		$b3_s$		$B3_s$				$w(b3_s)$	$u(b3_s)$	$U(b3_s)$		$W(b3_s)$	$F(b3_s)$			$F_N(b3_s)$	
Denial		$b3_\ell$		$B3_\ell$							$P(b3_\ell)$						
Rejected		$b3_f$		$B3_f$													
DISENGAGEMENT**	d		D		D′	t(d)											
Successful		d_s		D_s			$t(d_s)$	$w(d_s)$	$u(d_s)$	$U(d_s)$		$W(d_s)$	$F(d_s)$		$W_N(d_s)$	$F_N(d_s)$	
Denial		d_ℓ		D_ℓ							$P(d_\ell)$						
User Blocking		d_f		D_f													

*The "sample" defined here is essentially a unit of measure (see ANS X3.102).

**The symbol 1 or 2 is appended to the disengagement symbols d and D in each instance if source user disengagement (d1) and destination user disengagement (d2) are to be decribed with separate parameters.

performance time values for individual trials from the corresponding averages, denoted by W( ) and U( ), respectively.

(5) Specified parameter values are distinguished from the corresponding measured values by the uppercase subscript N. For example, the symbol $W_N(a_s)$ denotes the specified value for the parameter Access Time.

(6) Transfer Denial threshold values for supported performance parameters are identified by the uppercase subscript T. For example, the symbol $P_T(bl_e)$ denotes the transfer denial threshold for Bit Error Probability, i.e., the value of Bit Error Probability above which a Transfer Denial (or outage) is declared.

In all cases, the parameter estimates should be based on a reduced measurement sample that excludes failures attributable to the user. A prime (') symbol after a trial or outcome count identifies such a reduced measurement sample. For example, A' represents the number of access attempts in a measurement sample that excludes the A_f access failures attributable to user blocking.

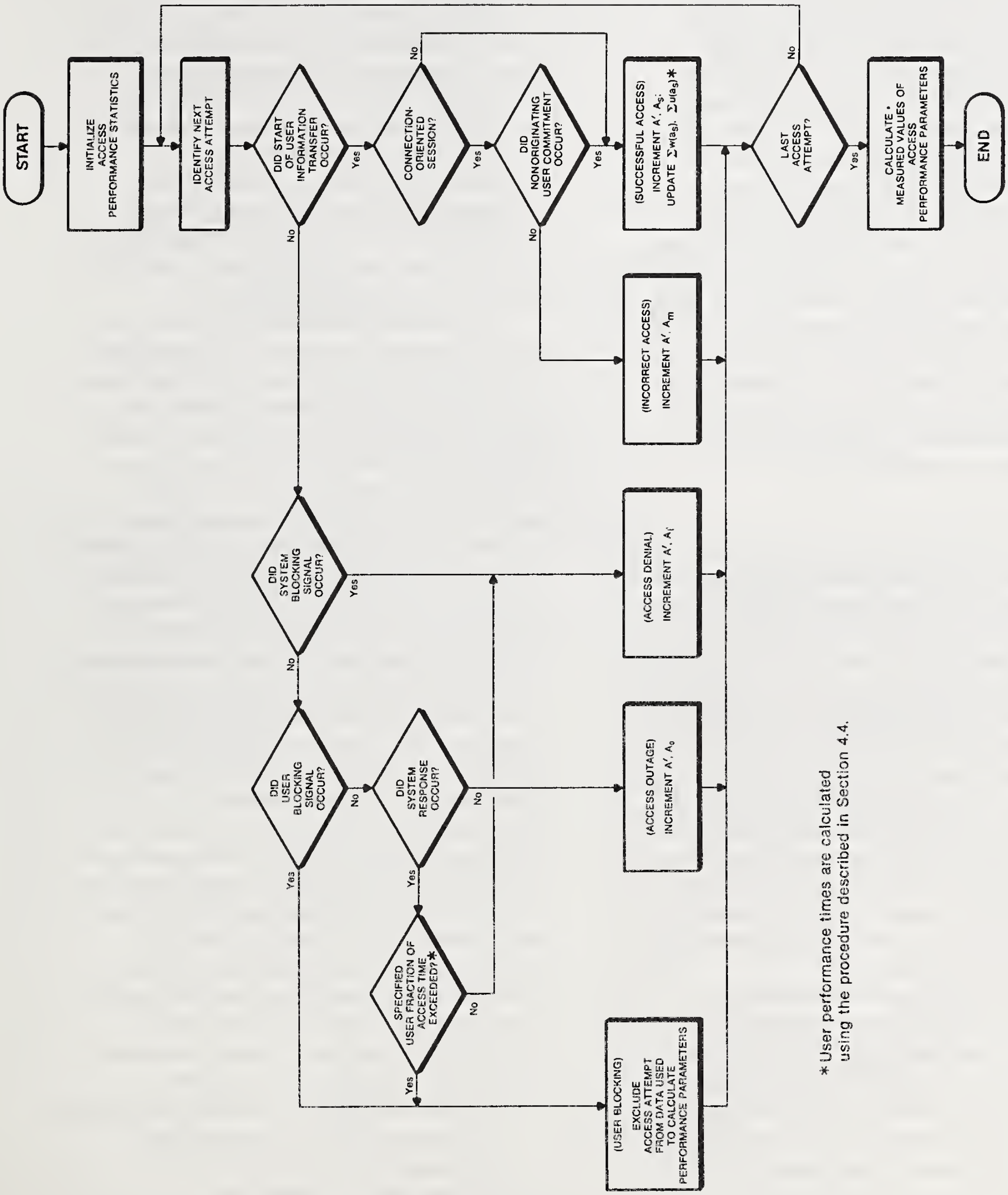
4.1 Access Parameter Calculation. Figure 7 defines a procedure for estimating the access parameter values. The procedure assumes as its input a sequence of recorded primary reference events representing the user/system interactions observed during a set of monitored access attempts. The procedure produces as its output an estimated value for each of the five access performance parameters. The user performance time calculation procedure (as described in 4.4) is used to estimate User Fraction of Access Time and to determine the entity responsible for access timeouts.

The first step in the access data reduction procedure is to initialize the variables used in recording the access outcomes. The beginning of each access attempt is identified via the Access Request event. For each access attempt in the measurement sample, a series of logical tests is conducted to determine which of five possible outcomes the attempt encountered: Successful Access, Incorrect Access, Access Denial, Access Outage, or User Blocking. The first test determines whether the attempt resulted in user information transfer, on the one hand, or blocking or timeout, on the other. This is determined by the presence or absence of a Start of Block Input to System event during the timeout period.

The processing of access attempts that result in user information transfer depends on whether the sessions in the sample are connection oriented or connectionless. In the connection-oriented case, a test for nonoriginating user commitment is made to determine whether the intended nonoriginating user was, in fact, contacted during (or prior to) the access attempt. If so, the attempt is placed in the Successful Access category; if not, the attempt is placed in the Incorrect Access category. Incorrect Access cannot occur in the connectionless case, and the attempt is therefore immediately placed in the Successful Access category.

Each time an attempt is placed in the Successful Access category, the number of Successful Access outcomes (A_s) is incremented by one; and the access time $w(a_s)$ and user access time $u(a_s)$ observed on that attempt are calculated and added to the corresponding cumulative totals, $\Sigma w(a_s)$ and $\Sigma u(a_s)$. The access time for a successful access attempt is the time difference between the Start of First Block Input and Access Request events, as discussed earlier. The user access time is determined by the procedure described in 4.4. Each time an access attempt is placed in the Incorrect Access category, the number of Incorrect Access outcomes (A_m) is incremented by one.

In cases in which access failure results from blocking or timeout, it is necessary to determine whether a user or the system was responsible; and in the latter case, whether the failure was an Access Denial (i.e., system blocking) or Access Outage. If the system issued a blocking signal during the access attempt, the attempt is immediately placed in the Access Denial category. If a user issued a blocking signal, the attempt is immediately placed in the User Blocking category. If no blocking signal was issued, the event history is checked to determine whether the system issued any other response during the access attempt. If not, the attempt is placed in the Access Outage category. Otherwise, the access attempt outcome is determined by calculating the user fraction of performance time for the particular (unsuccessful) trial in question (by means of the user performance time calculation procedure described in 4.4) and then comparing



* User performance times are calculated using the procedure described in Section 4.4.

Figure 7
Procedure for Estimating Access Parameter Values

the calculated value with the specified value for the ancillary parameter, User Fraction of Access Time. If the user fraction for the particular trial exceeds the specified value, the attempt is placed in the User Blocking category; otherwise, the failure is placed in the Access Denial category. Each time an attempt is placed in the Access Denial or Access Outage category, the corresponding outcome variable (A_l or A_o) is incremented by one. The total number of access trials (A') counted in assessing access performance is incremented on each Successful Access, Incorrect Access, Access Denial, or Access Outage outcome. User Blocking outcomes are not normally counted, since these are excluded from system performance measurement. After all access attempts in the measurement sample have been processed, estimates of the access performance parameters are calculated from the recorded data. Equations and associated definitions needed to estimate each access parameter are provided in Figure 8.

4.2 User Information Transfer Parameter Calculation. Figure 9 defines a procedure for estimating the user information transfer (UIT) parameter values. The procedure assumes as its input a sequence of recorded primary reference events (and associated user information records) representing the user/system interactions observed during a set of monitored block transfer attempts. The procedure produces as its output an estimated value for each of the 13 UIT performance parameters. The user performance time calculation procedure described in 4.4 is used to estimate the ancillary UIT parameters and to determine the entity responsible for block transfer or sample input/output timeouts.

The first step in the user information transfer data reduction procedure is to initialize the variables used in recording the UIT outcomes. The source and destination user information blocks are compared to match corresponding transmitted and received bits and to identify errored, undelivered, and extra bits. That "data correlation" process, illustrated schematically in Figure 10, produces a series of correlated output blocks recording individual source/destination Bit Comparison Outcomes (BCOs) in one of four categories defined as follows:

(1) *Correct BCO.* Corresponding bits exist in the source and destination user information records, and their binary values agree.

(2) *Incorrect BCO.* Corresponding bits exist in the source and destination user information records, but their binary values differ.

(3) *Undelivered BCO.* A bit in the source user information record has no counterpart in the destination user information record.

(4) *Extra BCO.* A bit in the destination user information record has no counterpart in the source user information record.

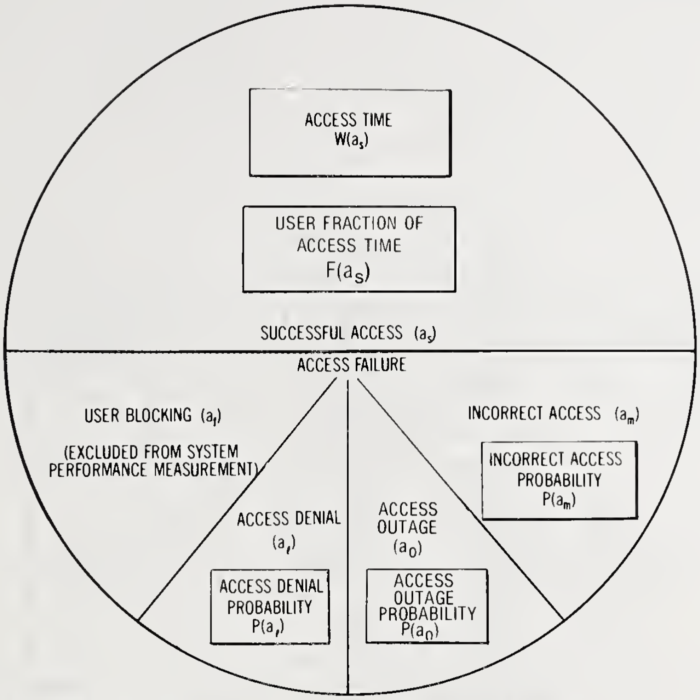
A subroutine that performs this data correlation function is included in the data reduction computer program mentioned earlier. Data correlator outputs containing each BCO type are illustrated in Figure 11.

After the data correlation processing has been completed, each correlated output block is examined (in conjunction with the associated transfer start and end times) to classify the bit comparison outcomes it contains. Blocks containing all undelivered BCOs and blocks that are "timed out" (i.e., have measured transfer times greater than the specified timeout value) are tested to determine whether the system or a user was responsible for the failure. This is determined by calculating the user fraction of performance time for the particular (unsuccessful) block transfer attempt in question (by means of the user performance time calculation procedure described in 4.4), and then comparing the calculated value with the specified value for the ancillary parameter, User Fraction of Block Transfer Time. If the user fraction for the particular trial exceeds the specified value, all BCOs in the correlated output block are classified as Refused Bits, and are excluded in calculating the UIT performance parameters. Otherwise (if the block timed out as a result of system delay), any extra BCOs are classified as Extra Bits and all other BCOs are classified as Lost Bits.

The BCOs in all other types of correlated output blocks are classified as follows:

(1) Correct BCOs = Successful Bit Transfers
(2) Incorrect BCOs = Incorrect Bits
(3) Undelivered BCOs = Lost Bits
(4) Extra BCOs = Extra Bits

The outcome variables associated with these bit transfer outcomes ($B1_s$, $B1_f$, $B1_l$, $B1_e$, $B1_x$, and B1') are updated after all bit comparison outcomes in a correlated output block have been classified. The recorded bit transfer



ACCESS PARAMETER EQUATIONS

1. Access Time $= W(a_s) = \frac{1}{A_s} \sum_{a_s=1}^{A_s} w(a_s)$
2. Incorrect Access Probability $= P(a_m) = A_m/A'$
3. Access Denial Probability $= P(a_\ell) = A_\ell/A'$
4. Access Outage Probability $= P(A_o) = A_o/A'$
5. User Fraction of Access Time $= F(a_s) = U(a_s)/W(a_s)$

DEFINITIONS

A' = Total number of access attempts counted during an access parameter measurement: $A_s + A_m + A_\ell + A_o$

A_s = Total number of Successful Access outcomes counted during an access parameter measurement.

A_ℓ = Total number of Access Denial outcomes counted during an access parameter measurement.

A_o = Total number of Access Outage outcomes counted during an access parameter measurement.

A_m = Total number of Incorrect Access outcomes counted during an access parameter measurement.

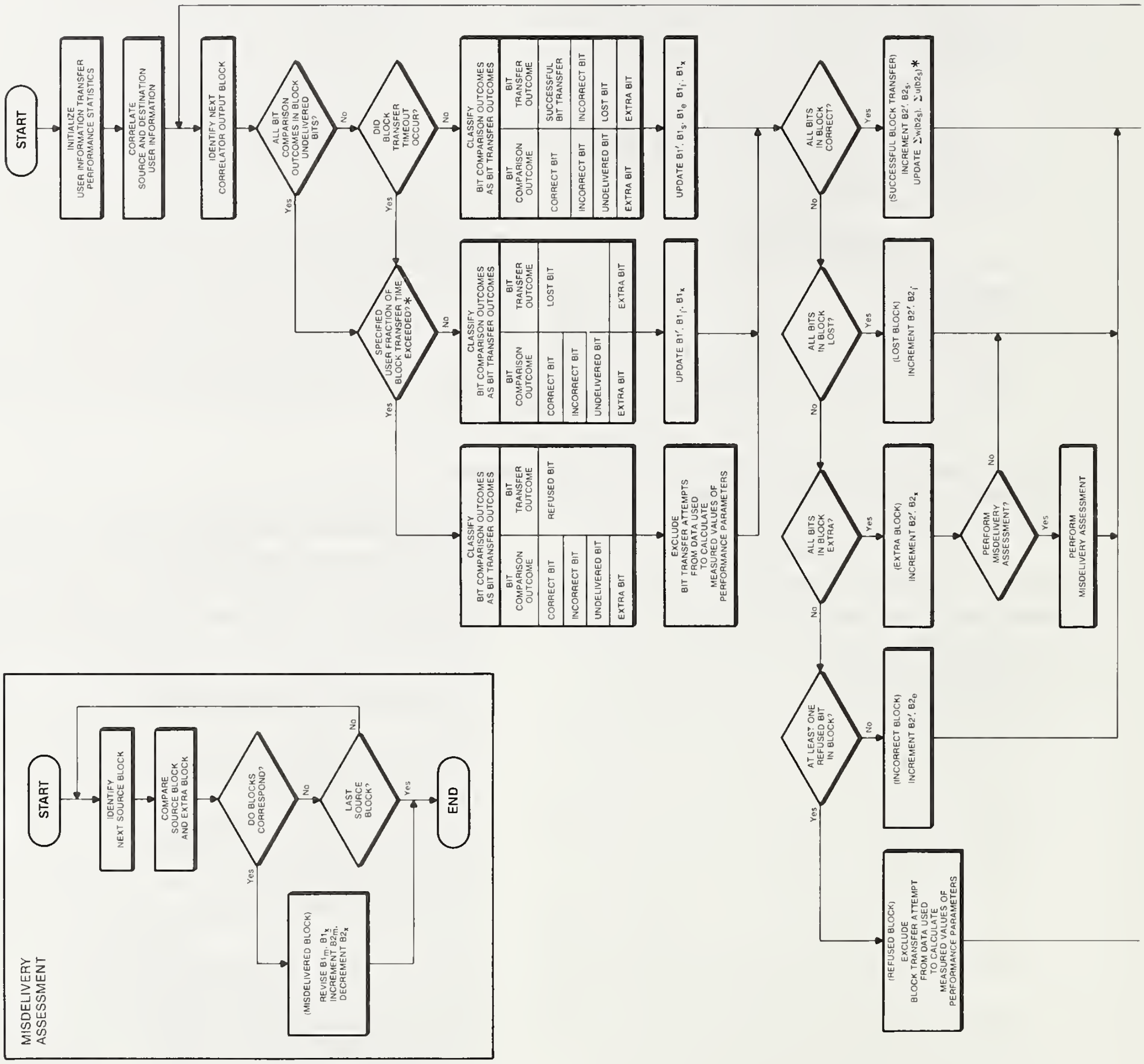
$w(a_s)$ = Value of access time measured on a particular successful access attempt.

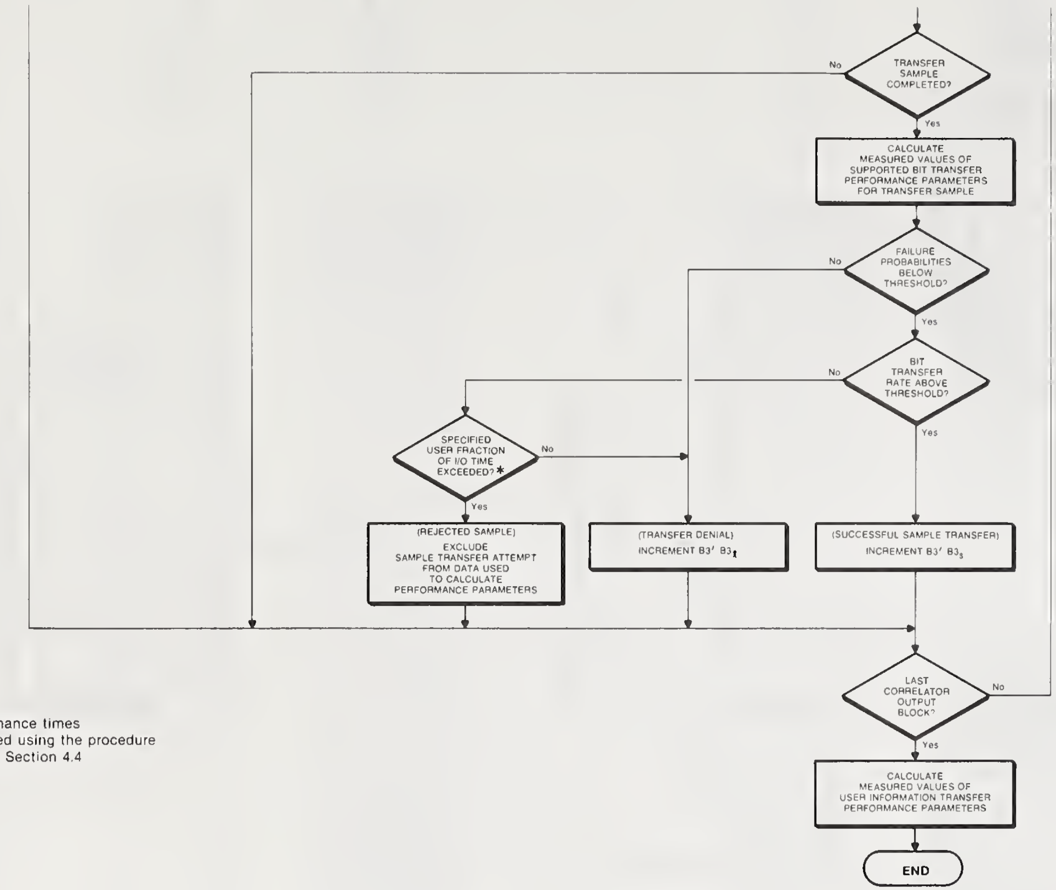
$u(a_s)$ = Value of user access time measured on a particular successful access attempt.

$U(a_s)$ = Average user access time measured over A_s successful access attempts

$$= \frac{1}{A_s} \sum_{a_s=1}^{A_s} u(a_s)$$

Figure 8
Access Parameter Definitions


START
INITIALIZE USER INFORMATION TRANSFER PERFORMANCE STATISTICS
CORRELATE SOURCE AND DESTINATION USER INFORMATION
IDENTIFY NEXT CORRELATOR OUTPUT BLOCK
ALL BIT COMPARISON OUTCOMES IN BLOCK UNDELIVERED BITS?
DID BLOCK TRANSFER TIMEOUT OCCUR?
SPECIFIED USER FRACTION OF BLOCK TRANSFER TIME EXCEEDED?*
CLASSIFY BIT COMPARISON OUTCOMES AS BIT TRANSFER OUTCOMES
BIT COMPARISON OUTCOME
BIT TRANSFER OUTCOME
CORRECT BIT
SUCCESSFUL BIT TRANSFER
INCORRECT BIT
UNDELIVERED BIT
LOST BIT
EXTRA BIT
REFUSED BIT
UPDATE B1', B1s, B1e, B1l, B1x
UPDATE B1', B1l, B1x
EXCLUDE BIT TRANSFER ATTEMPTS FROM DATA USED TO CALCULATE MEASURED VALUES OF PERFORMANCE PARAMETERS
ALL BITS IN BLOCK CORRECT?
(SUCCESSFUL BLOCK TRANSFER) INCREMENT B2', B2s. UPDATE Σw(b2s), Σu(b2s)*
ALL BITS IN BLOCK LOST?
(LOST BLOCK) INCREMENT B2', B2l
ALL BITS IN BLOCK EXTRA?
(EXTRA BLOCK) INCREMENT B2', B2x
PERFORM MISDELIVERY ASSESSMENT?
PERFORM MISDELIVERY ASSESSMENT
AT LEAST ONE REFUSED BIT IN BLOCK?
(INCORRECT BLOCK) INCREMENT B2', B2e
(REFUSED BLOCK) EXCLUDE BLOCK TRANSFER ATTEMPT FROM DATA USED TO CALCULATE MEASURED VALUES OF PERFORMANCE PARAMETERS
Yes
No
MISDELIVERY ASSESSMENT
START
IDENTIFY NEXT SOURCE BLOCK
COMPARE SOURCE BLOCK AND EXTRA BLOCK
DO BLOCKS CORRESPOND?
LAST SOURCE BLOCK?
END
(MISDELIVERED BLOCK) REVISE B1m, B1x INCREMENT B2m, DECREMENT B2x



* User performance times are calculated using the procedure described in Section 4.4

Figure 9
Procedure for Estimating User Information Transfer Parameter Values

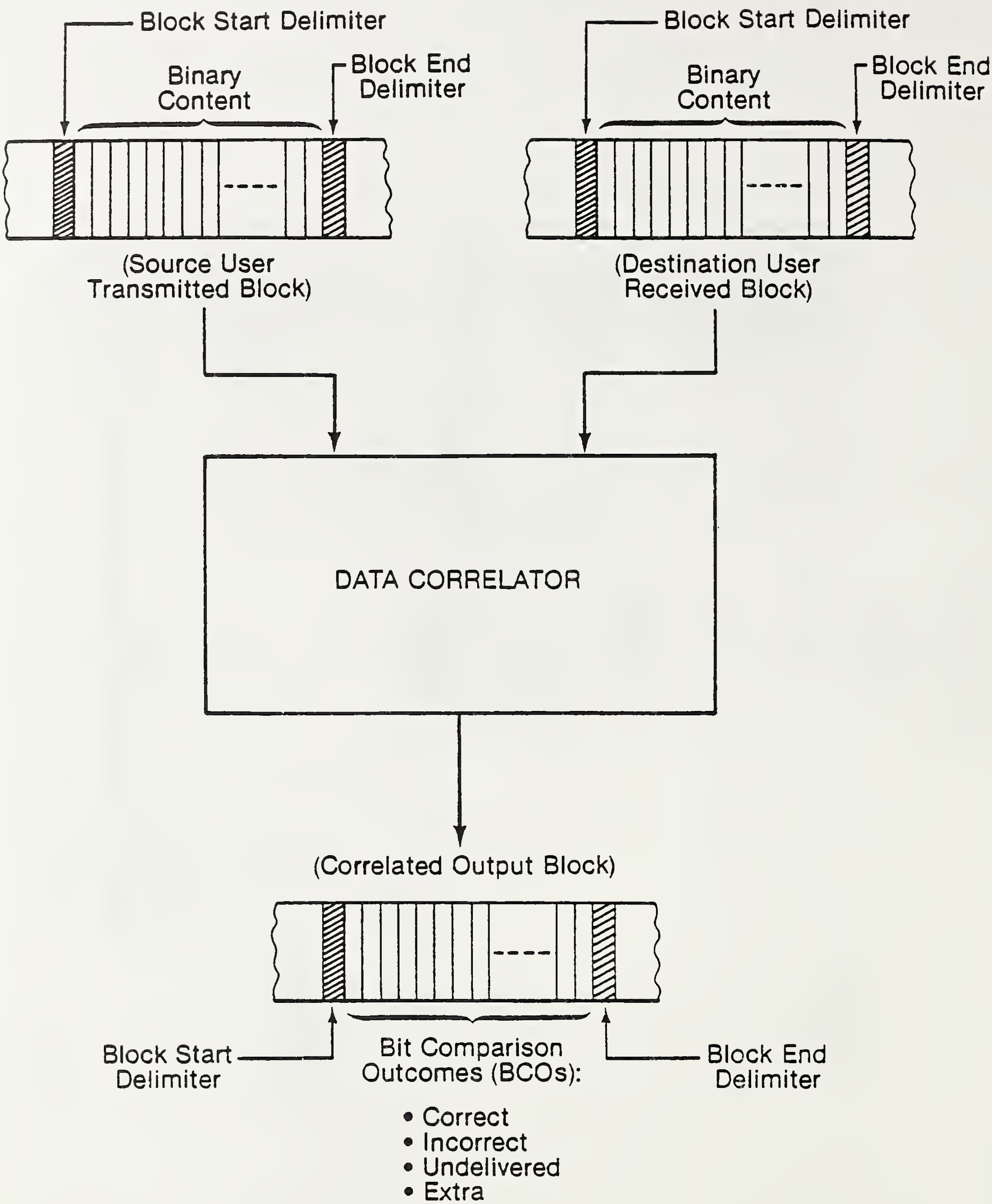
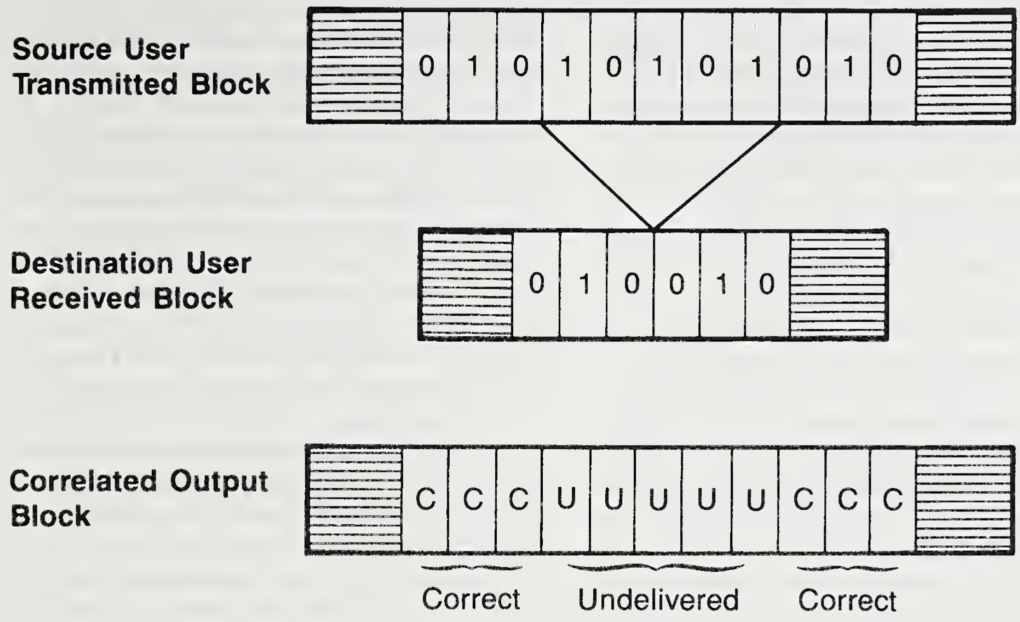
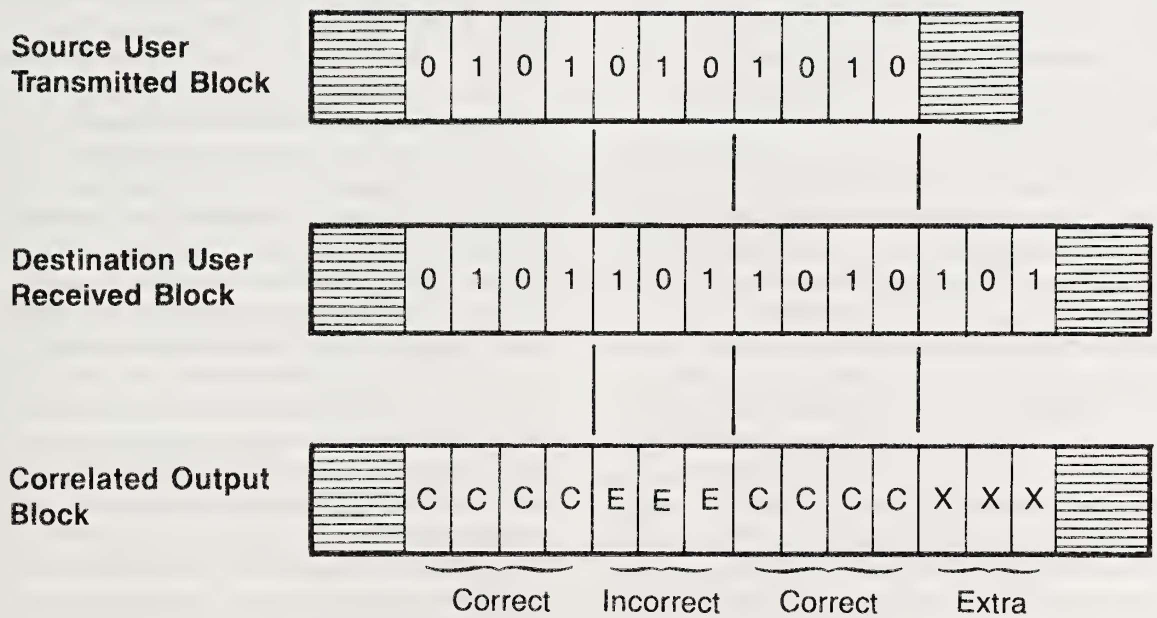


Figure 10
Schematic Representation of the Data Correlation Process



(*a*) *Example 1: Undelivered Bits within a Block*



(*b*) *Example 2: Incorrect and Extra Bits within a Block*

Figure 11
Examples of Data Correlator Outputs

outcomes are then used to define a transfer outcome for the overall block. If all bits in a correlated output block are classified as Successful Bit Transfers, the block transfer outcome is defined as Successful Block Transfer. If all bits are classified as Lost Bits, the block is classified as a Lost Block. If all bits are classified as Extra Bits, the block is classified as an Extra Block. If one or more bits in a correlated output block are classified as Refused bits, the block is classified as a Refused Block. Correlated blocks containing any other combination of bit transfer outcomes (e.g., blocks containing both Successful Bit Transfers and Incorrect Bits) are classified as Incorrect Blocks.

Each time an attempt is placed in the Successful Block Transfer, Lost Block, Extra Block, or Incorrect Block category, the corresponding block transfer outcome variable ($B2_s$, $B2_l$, $B2_x$, or $B2_e$) and the total number of block transfer trials (B2') are incremented by one. In the case of Successful Block Transfers, the block transfer time $w(b2_s)$ and user block transfer time $u(b2_s)$ are calculated and added to the corresponding cumulative totals, $\Sigma w(b2_s)$ and $\Sigma u(b2_s)$. The block transfer time for a successful block transfer attempt is the difference between the End of Block Transfer and Start of Block Transfer events, as discussed earlier. The user block transfer time is determined by the procedure described in 4.4.

The additional processing needed to distinguish Misdelivered Blocks from Extra Blocks is illustrated in the inset at the upper left of Figure 9. Misdelivered Blocks are identified by comparing each Extra Block observed at the monitored destination user interface with any blocks the monitored source user transmitted to the system *for delivery to other destination users* during the measurement period. Extra Blocks that can be associated with blocks transmitted to other destination users are reclassified as Misdelivered Blocks, and the associated bits are reclassified as Misdelivered Bits. This process is optional and will often be omitted, as discussed in ANSI X3.102-1983.

The lower portion of Figure 9 illustrates the process used in estimating values for Transfer Denial Probability, a sampled measure of unavailability. As defined in ANSI X3.102-1983, a **transfer sample** is a selected observation of user information transfer performance between a specified source and destination user. A transfer sample includes an integer number of user information blocks, and the interblock gap that precedes each block. The size of the transfer sample shall be selected by the test operator to provide estimates of known precision for four supported performance parameters: Bit Error Probability, Bit Loss Probability, Extra Bit Probability, and User Information Bit Transfer Rate. This is discussed in ANSI X3.102-1983. The Transfer Denial Thresholds for the four parameters are defined as follows:

(1) The Transfer Denial threshold for each of the three bit transfer failure probabilities is defined as the fourth root of the probability value specified for the service. For example, the Transfer Denial threshold for a service with a specified Bit Error Probability of 10^{-8} would be $10^{-8/4}$, or 10^{-2}.

(2) The Transfer Denial threshold for User Information Bit Transfer Rate is defined as one-third of the User Information Bit Transfer Rate specified for the service. For example, the Transfer Denial threshold for a service with a specified User Information Bit Transfer Rate of 2400 bits per second (bps) would be 2400/3 = 800 bps.

The Transfer Denial Probability estimation procedure illustrated in Figure 9 divides all UIT performance data recorded during a measurement into a succession of transfer samples, and determines the outcome of each transfer sample using the criteria defined in this subsection. As shown in the figure, each correlated output block is checked to determine whether it is the last block in a transfer sample. If a particular block is not the last in a transfer sample, no transfer sample processing is required. Otherwise, values for the four supported performance parameters are calculated over the transfer sample, based on recorded outcomes and event times. Each calculated failure probability value is compared with its associated threshold. If any calculated probability is above its threshold value, the transfer sample is placed in the Transfer Denial category. If all three calculated values are at or below their respective thresholds, the calculated User Information Bit Transfer Rate is compared with the threshold rate. If the calculated rate equals or exceeds the threshold rate, the transfer sample is placed in the Successful Sample Transfer category. Otherwise, it is necessary to determine whether a user or the system was responsible for the failure. This is done by calculating the user fraction of input/output time for the particular (unsuccessful) sample in question (by means of the user performance time calculation procedure described in 4.4) and then comparing the calculated value with the specified value for the ancillary parameter User Fraction of Input/Output Time. If the user fraction for the particular trial exceeds the specified value, the transfer sample is classified as a Rejected Sample and is excluded in calculating Transfer Denial Probability. Otherwise, the transfer sample is placed in the Transfer Denial category. The appropriate outcome variable ($B3_s$ or $B3_l$) is incremented by one each time a sample transfer outcome is determined.

After all transfer attempts in a UIT measurement have been processed, the long-term UIT performance parameter values are estimated from the complete set of recorded data. The duration of the measurement should be sufficient to ensure that the desired measurement precision is achieved [4]. In a measurement of long-term steady-state throughput, the input and output times considered should be equal. The reported User Fraction of Input/Output Time should be the larger of the observed user fractions. Equations and associated definitions needed to calculate each UIT parameter are provided in Figures 12 through 14.

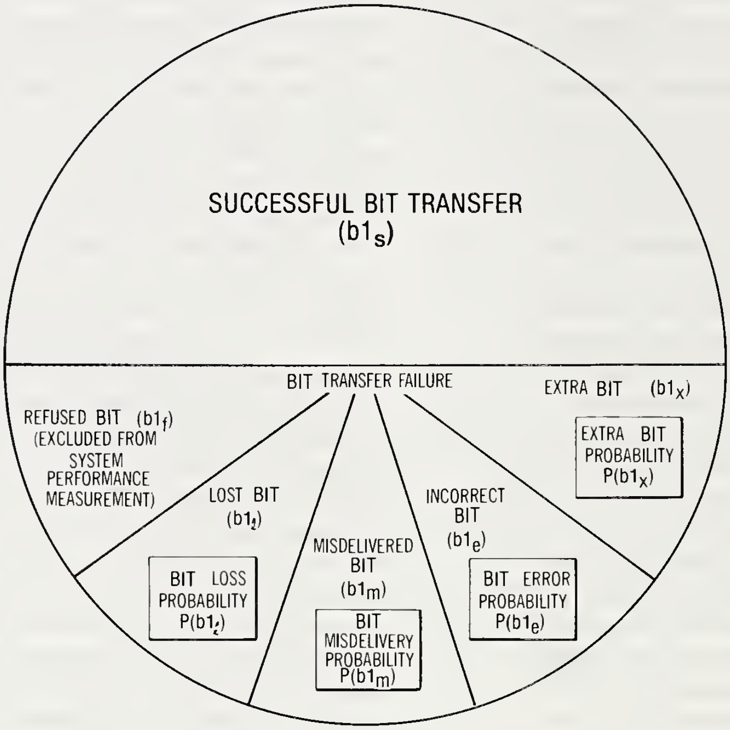
4.3 Disengagement Parameter Calculation. Figure 15 defines a procedure for estimating the disengagement parameter values. The procedure assumes as its input a sequence of recorded primary reference events representing the user/system interactions during a set of monitored disengagement attempts. The procedure produces as its output an estimated value for each of the three disengagement performance parameters. The user performance time calculation procedure (see 4.4) is used to estimate the User Fraction of Disengagement Time and to determine the entity responsible for disengagement timeouts.

A typical data communication session involves two disengagement functions: one for each participating user. In defining the disengagement parameters in ANSI X3.102-1983, two options are identified: (1) calculate separate disengagement parameters for each user and (2) combine the disengagement outcomes of both users to calculate parameters representing the "average" disengagement performance. The former procedure is assumed here, since the latter is a straightforward simplification of it. The symbols d1 and d2 represent the source user and destination user disengagement functions, respectively.

The first step in the disengagement data reduction procedure is to initialize the variables used in recording the disengagement outcomes. The beginning of each disengagement attempt is identified via a Disengagement Request event. In the case of connection-oriented services, the first Disengagement Request after Successful Access is interpreted as a request to disengage both users. In the case of connectionless services, a separate Disengagement Request is identified at each user interface. In either case, the outcomes of the source user and destination user disengagement attempts are determined independently. The possible outcomes are Successful Disengagement, Disengagement Denial, and User Disengagement Blocking.

On any given attempt, Successful Disengagement is indicated by the occurrence of Disengagement Confirmation at the appropriate user/system interface within the timeout period. Each time an attempt is placed in the Successful Disengagement category, the appropriate Successful Disengagement counter ($D1_s$ or $D2_s$) is incremented by one; the disengagement time, $w(d1_s)$ or $w(d2_s)$, and user disengagement time, $u(d1_s)$ or $u(d2_s)$, observed on that attempt are calculated; and each observed time is added to its corresponding cumulative total, $\Sigma w(d1_s)$, $\Sigma w(d2_s)$, $\Sigma u(d1_s)$ or $\Sigma u(d2_s)$. The disengagement time for a successful disengagement attempt is the time difference between the Disengagement Confirmation and Disengagement Request events, as discussed earlier. The user disengagement time is determined by the procedure described in 4.4.

On any given attempt, the absence of Disengagement Confirmation within the timeout period indicates disengagement failure. When such outcomes occur, it is necessary to determine whether a user or the system was responsible. This is determined by calculating



BIT TRANSFER PARAMETER EQUATIONS

1. Bit Loss Probability $= P(b1_\ell) = B1_\ell / (B1_s + B1_e + B1_\ell)$
2. Bit Misdelivery Probability $= P(b1_m) = B1_m / (B1' - B1_\ell - B1_x)$
3. Bit Error Probability $= P(b1_e) = B1_e / (B1_s + B1_e)$
4. Extra Bit Probability $= P(b1_x) = B1_x / (B1' - B1_\ell)$

DEFINITIONS

$B1'$ = Total number of bit transfer outcomes to be included in an individual UIT performance measurement (all bit transfer outcomes except $b1_f$).

$B1_s$ = Total number of Successful Bit Transfer outcomes counted during a UIT performance measurement.

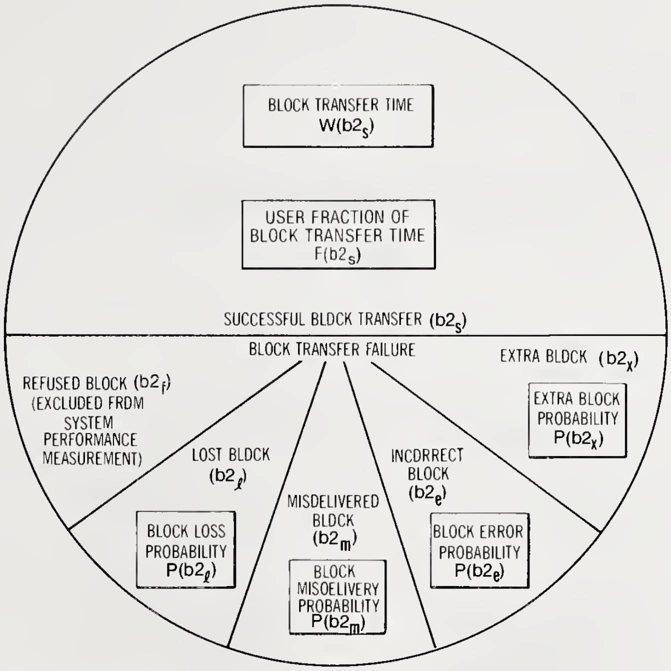
$B1_\ell$ = Total number of Lost Bit outcomes counted during a UIT performance measurement.

$B1_m$ = Total number of Misdelivered Bit outcomes counted during a UIT performance measurement.

$B1_e$ = Total number of Incorrect Bit outcomes counted during a UIT performance measurement.

$B1_x$ = Total number of Extra Bit outcomes counted during a UIT performance measurement.

Figure 12
User Information Bit Transfer Parameter Definitions



BLOCK TRANSFER PARAMETER EQUATIONS

1. Block Transfer Time = $W(b2_s) = \frac{1}{B2_s} \sum_{b2_s = 1}^{B2_s} w(b2_s)$
2. Block Loss Probability = $P(b2_\ell) = B2_\ell / (B2_s + B2_e + B2_\ell)$
3. Block Misdelivery Probability = $P(b2_m) = B2_m/(B2' - B2_\ell - B2_x)$
4. Block Error Probability = $P(b2_e) = B2_e/(B2_s + B2_e)$
5. Extra Block Probability = $P(b2_x) = B2_x/(B2' - B2_\ell)$
6. User Fraction of Block Transfer Time = $F(b2_s) = U(b2_s)/W(b2_s)$

DEFINITIONS

$B2'$ = Total number of block transfer outcomes to be included in an individual UIT performance measurement (all block transfer outcomes except $b2_r$).

$B2_s$ = Total number of Successful Block Transfer outcomes counted during a UIT performance measurement.

$B2_\ell$ = Total number of Lost Block outcomes counted during a UIT performance measurement.

$B2_m$ = Total number of Misdelivered Block outcomes counted during a UIT performance measurement.

$B2_e$ = Total number of Incorrect Block outcomes counted during a UIT performance measurement.

$B2_x$ = Total number of Extra Block outcomes counted during a UIT performance measurement.

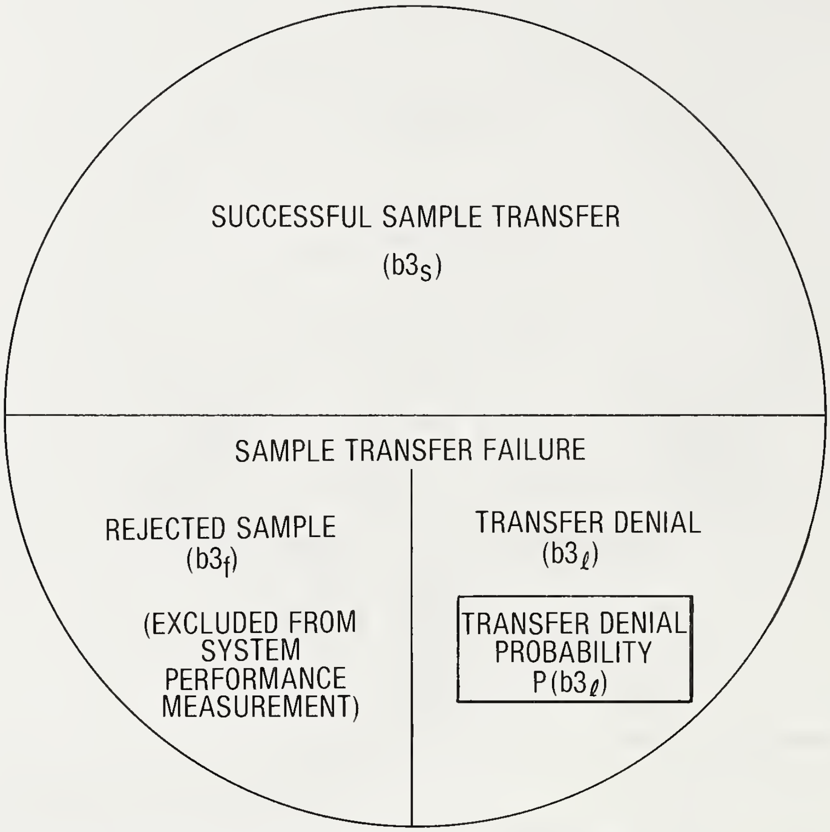
$w(b2_s)$ = Value of block transfer time measured on a particular successful block transfer attempt.

$u(b2_s)$ = Value of user block transfer time measured on a particular successful block transfer attempt.

$U(b2_s)$ = Average user block transfer time measured over $B2_s$ successful block transfer attempts

$$= \frac{1}{B2_s} \sum_{b2_s = 1}^{B2_s} u(b2_s).$$

Figure 13
User Information Block Transfer Parameter Definitions



SAMPLE TRANSFER PARAMETER EQUATIONS

1. Transfer Denial Probability = $P(b3_\ell) = B3_\ell / B3'$
2. User Information Bit Transfer Rate = $R(b1_s) = B1_s/w(b3)$
3. User Fraction of Input/Output Time = $F(b3) = u(b3)/w(b3)$

DEFINITIONS

$B3'$ = Total number of transfer samples to be included in a Transfer Denial Probability determination.

$B3_\ell$ = Total number of Transfer Denial outcomes counted in a Transfer Denial Probability determination.

$B1_s$ = Total number of Successful Bit Transfer outcomes.

$w(b3)$ = Greater of input time or output time measured in a particular observation.

$u(b3)$ = Value of user input/output time measured in a particular observation.

In the case where the input time and the output time are equal, the user input/output time is the larger of the user input time and the user output time.

Figure 14
Sample Transfer Parameter Definitions

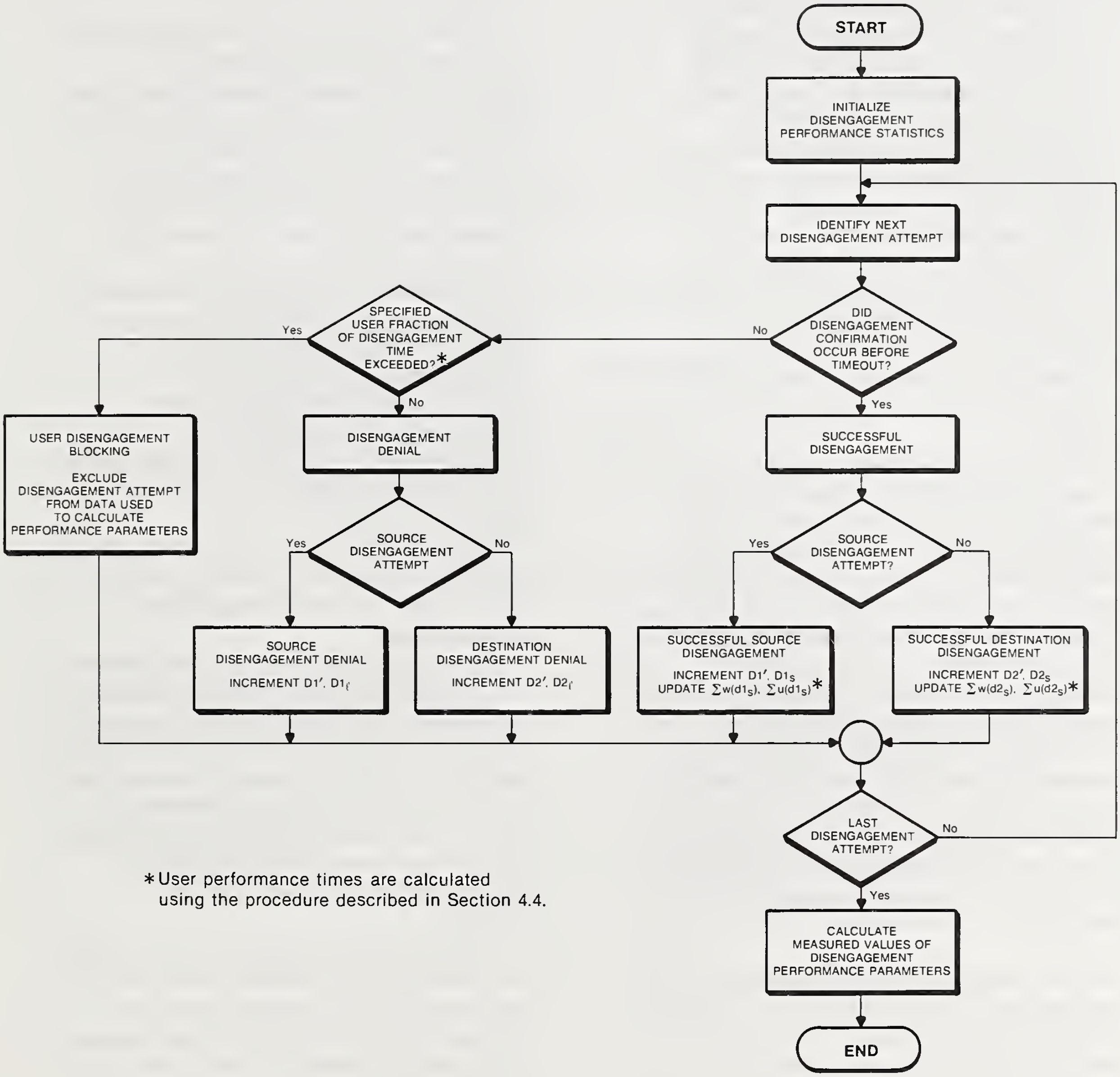


Figure 15
Procedure for Estimating Disengagement Parameter Values

the user fraction of performance time for the particular (unsuccessful) trial in question (making use of the user performance time calculation procedure described in 4.4) and then comparing the calculated value with the specified value for the ancillary parameter User Fraction of Disengagement Time. If the user fraction for the particular trial exceeds the specified value, the attempt is placed in the User Disengagement Blocking category; otherwise, the attempt is placed in the Disengagement Denial category. Each time an attempt is placed in the Disengagement Denial category, the appropriate outcome variable ($D1_l$ or $D2_l$) is incremented by one. The appropriate total number of disengagement trials (D1' or D2') is incremented on each Successful Disengagement and Disengagement Denial outcome. User Disengagement Blocking outcomes are not normally counted, since these are excluded from system performance measurement.

After all disengagement attempts in the measurement sample have been processed, the disengagement performance parameters are calculated from the recorded data. Equations and associated definitions needed to calculate each disengagement parameter are provided in Figure 16. General disengagement symbols are used for brevity (e.g., d_s for Successful Disengagement). The disengagement equations apply as presented in cases in which the source and destination user disengagement outcomes are combined to calculate average performance values. Each equation can be specialized to a particular user by appending 1 or 2 to the corresponding user-general symbol (e.g., $d1_s$ for Successful Disengagement of the source user).

4.4 User Performance Time Calculation. This subsection specifies a procedure for processing recorded ancillary reference events to determine, for a particular primary function performance period, the total performance time that is attributable to user delays. The procedure requires two basic types of inputs:

(1) An ancillary event history recorded at each of the monitored user/system interfaces. Each ancillary event record should include (a) the event time, (b) the subsequent responsibility state (user responsible, system responsible, or responsibility undefined) at the local (monitored) interface, and (c) the presence/absence of a responsibility effect at the remote interface. The latter two items may be specified using the ancillary event numbers defined in Table 5. Each ancillary event history should also begin with the responsibility state at the local interface prior to the first event.

(2) Specifications (input from the invoking access, user information transfer, or disengagement assessment procedure) that define (a) the starting and ending times of the period in which user performance time is to be determined and (b) the particular interface or interfaces that are relevant (i.e., should be examined) in defining overall responsibility during that period.

The user performance times calculated by this procedure are utilized by the access, user information transfer, and disengagement assessment procedures in two ways:

(1) To calculate estimated values of ancillary performance parameters

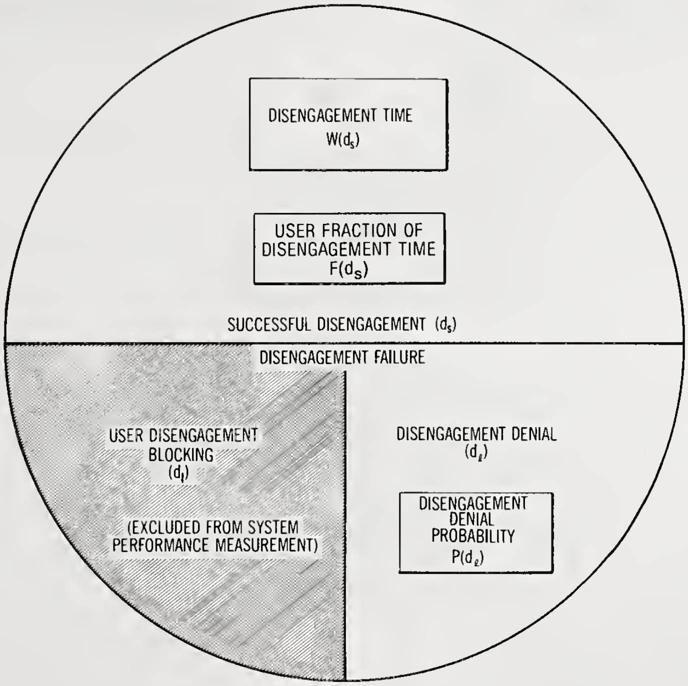
(2) To assign responsibility for performance timeout failures to either the system or the users

A summary of the user performance time calculation procedure is presented in Figure 17. The procedure consists of two principal subprocedures:

(1) An *event history consolidation* procedure that processes the ancillary events recorded by the source and destination interface monitors during a performance measurement period, and produces a unified and comprehensive ancillary event history for that period. Each ancillary event record output by this procedure includes (a) the event time and (b) the subsequent responsibility states at both monitored interfaces.

(2) A *performance time allocation* procedure that examines the consolidated ancillary event history to identify intervals of overall user responsibility within the specified performance period and determines total user performance time during that period.

The ancillary events recorded by a particular interface monitor do not, in general, provide a complete history of the responsibility states at the monitored interface, since responsibility at that interface can be affected by events at the remote interface. Collectively, however, the ancillary events recorded by the source and destination interface monitors in a performance measurement contain



DISENGAGEMENT PARAMETER EQUATIONS

1. Disengagement Time $= W(d_s) = \frac{1}{D_s} \sum_{d_s=1}^{D_s} w(d_s)$
2. Disengagement Denial Probability $= P(d_\ell) = D_\ell / D'$
3. User Fraction of Disengagement Time $= F(d_s) = U(d_s)/W(d_s)$

DEFINITIONS

D' = Total number of disengagement attempts counted during a disengagement parameter measurement: $D_s + D_\ell$.

D_s = Total number of Successful Disengagement outcomes counted during a disengagement parameter measurement.

D_ℓ = Total number of Disengagement Denial outcomes counted during a disengagement parameter measurement.

$w(d_s)$ = Value of disengagement time measured on a particular successful disengagement attempt.

$u(d_s)$ = Value of user disengagement time measured on a particular successful disengagement attempt.

$U(d_s)$ = Average user disengagement time measured over D_s successful disengagement attempts

$$= \frac{1}{D_s} \sum_{d_s=1}^{D_s} u(d_s).$$

Figure 16
Disengagement Parameter Definitions

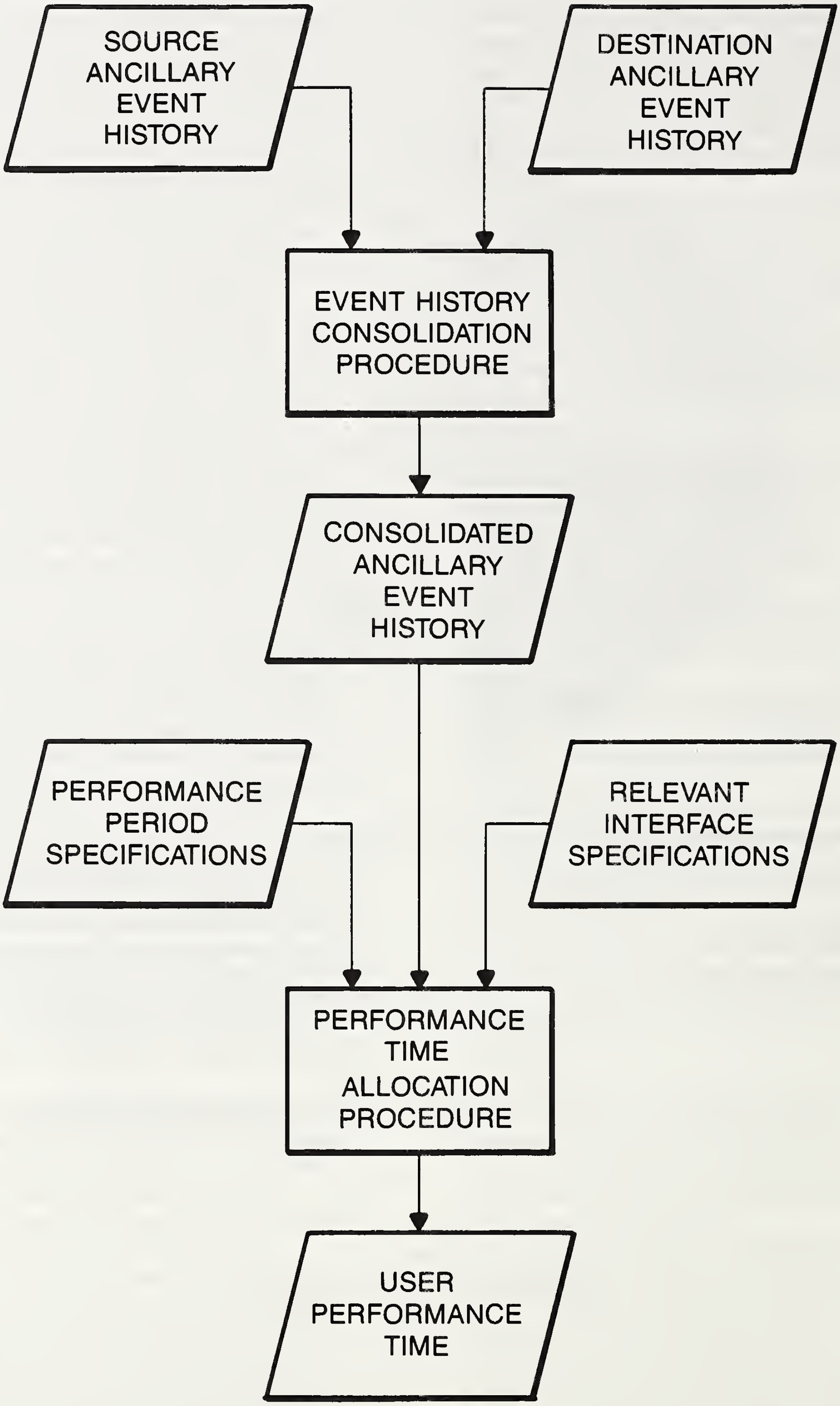


Figure 17
User Peformance Time Calculation Procedure

sufficient information to determine the complete responsibility state history at both interfaces. The event history consolidation procedure produces this complete history.

The individual interface responsibility states generated by the event history consolidation procedure are derived from two basic rules:

(1) An ancillary event at a particular interface always determines the subsequent responsibility state at that interface: user responsible, system responsible, or responsibility undefined.

(2) An ancillary event at a particular interface affects responsibility at the remote interface only if both entities at the remote interface are waiting for that event (i.e., if the responsibility state at the remote interface is "responsibility undefined"). In all such cases, the responsibility state at the remote interface is changed from "responsibility undefined" to "system responsible."

Figure 18 presents a functional flowchart for the event history consolidation procedure. The procedure begins by initializing the responsibility state at each interface to the state existing prior to the first ancillary event recorded at the respective interface. It next identifies the earliest event included in the unprocessed source and destination ancillary events, and determines the responsibility states at the local and remote interfaces in accordance with the basic rules specified previously. This procedure continues until all ancillary events have been processed.

Figure 19 illustrates the basic concepts used in the performance time allocation procedure. Each performance period begins with a primary reference event and ends with either a primary reference event or a timeout. The performance period is divided into a sequence of responsibility intervals by the ancillary events included in the period. Corresponding to each interval is an *overall responsibility state*, which is based on the responsibility state at the interface or interfaces relevant for the specified performance period. With the possible exception of the first and last intervals, each overall responsibility interval in a performance period is bounded by a pair of successive ancillary events. The beginning of the first interval and the end of the last interval are defined by the beginning and end, respectively, of the specified performance period. The user performance time in any particular performance period is the sum of the durations of the intervals of overall user responsibility within that period.

User performance time may be calculated for any of four types of performance periods:

(1) The period between the beginning and end of an access attempt

(2) The period between the beginning and end of a block transfer attempt

(3) The period delimiting the larger of the input time or the output time for an individual transfer sample or an overall UIT measurement

(4) The period between the beginning and end of a source user or destination user disengagement attempt

Table 7 defines the specific interface or interfaces that are relevant in determining user performance time for each type of performance period. Terms used in the table shall be as defined in ANSI X3.102 with the exception of those describing the conditions of disengagement. A **negotiated disengagement attempt** is one whose successful completion requires a concurring response from the user not initiating the disengagement request. An **independent disengagement attempt** is one that does not.

When only one interface is relevant in a performance period, the overall responsibility state for a particular responsibility interval is identical to the responsibility state at the relevant interface, as recorded in the consolidated event history. When both monitored interfaces are relevant, overall responsibility states are defined jointly by the two interface responsibility states in accordance with to the scheme presented in Table 8.

Table 8 includes "split" responsibility states, in which the user is responsible at one interface and the system is responsible at the other. If such intervals are contained in a performance period, their effect on user performance time is correctly accounted for by including them in the earliest subsequent responsibility interval that is not "split." Thus, if a user and the system simultaneously delay the completion of a function, responsibility for the joint delay is attributed to whichever entity delays longer.

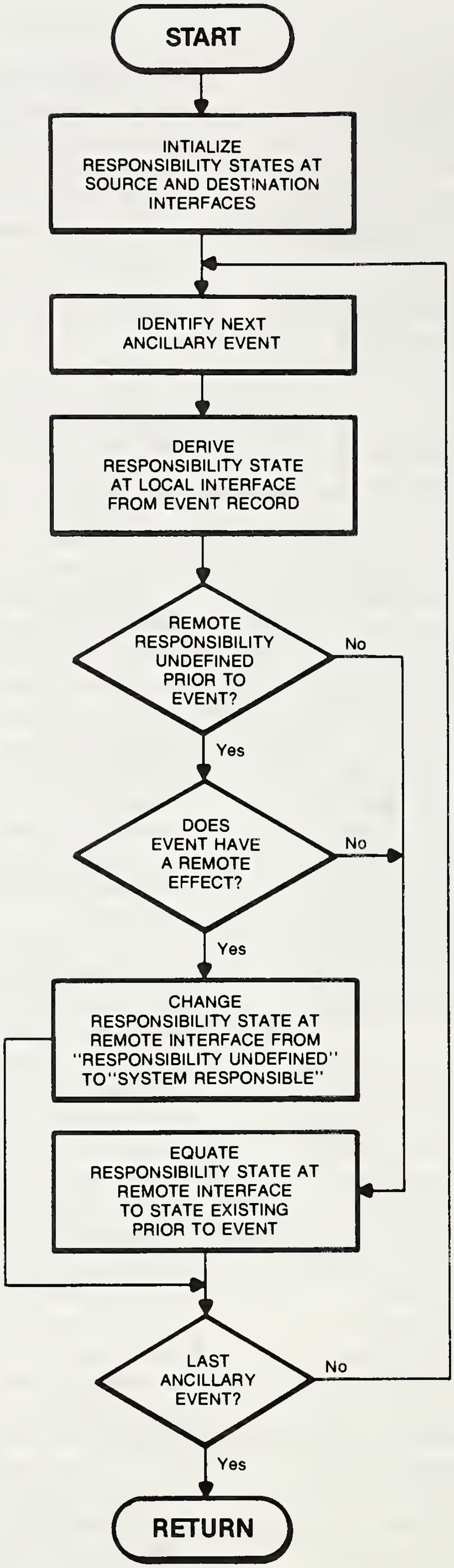
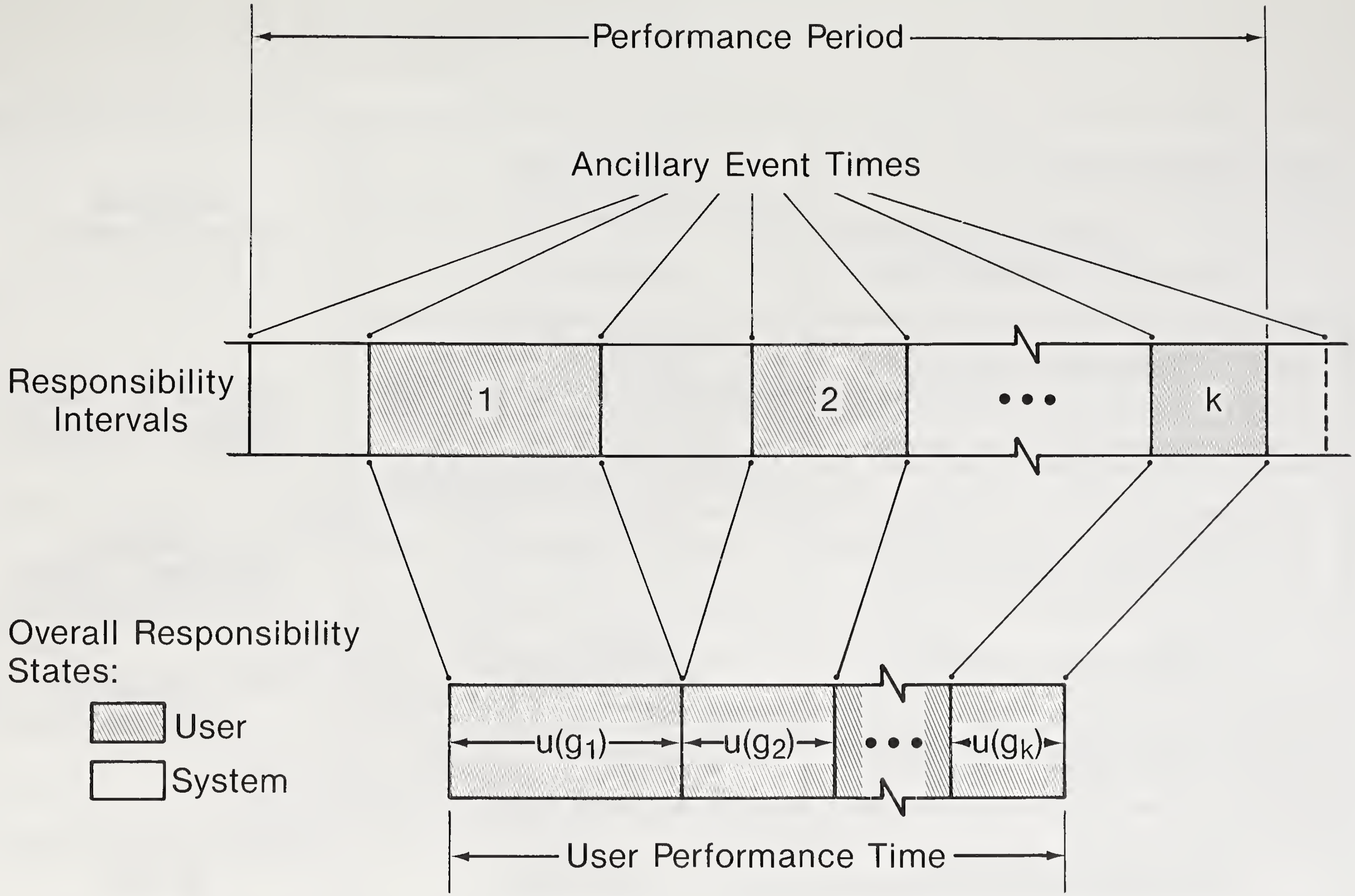


Figure 18 – Event History Consolidation Procedure



$$u(g) = \sum_{i=1}^{k} u(g_i)$$

Figure 19
Performance Time Allocation Concepts and Notation

Table 7
Relevant Interfaces for Performance Time Allocation

TYPE OF PERFORMANCE PERIOD	CONDITIONS	RELEVANT INTERFACES
ACCESS ATTEMPT	CONNECTION-ORIENTED	SOURCE AND DESTINATION USER
	CONNECTIONLESS	SOURCE USER ONLY
BLOCK TRANSFER ATTEMPT	ALL	DESTINATION USER ONLY
TRANSFER SAMPLE INPUT OR OUTPUT	RESPONSIBILITY DEFINED AT LOCAL INTERFACE	LOCAL USER ONLY
	RESPONSIBILITY UNDEFINED AT LOCAL INTERFACE	REMOTE USER ONLY
DISENGAGEMENT ATTEMPT	INDEPENDENT	REQUESTING USER ONLY
	NEGOTIATED	SOURCE AND DESTINATION USER

Table 8
Overall Responsibility States

Local Responsiblity State at Destination Interface \ Local Responsibility State at Source Interface	USER RESPONSIBLE	SYSTEM RESPONSIBLE	RESPONSIBILITY UNDEFINED
USER RESPONSIBLE	USER RESPONSIBLE	"SPLIT" RESPONSIBLITY	USER RESPONSIBLE
SYSTEM RESPONSIBLE	"SPLIT" RESPONSIBILITY	SYSTEM RESPONSIBLE	SYSTEM RESPONSIBLE
RESPONSIBILITY UNDEFINED	USER RESPONSIBLE	SYSTEM RESPONSIBLE	

Overall Responsibility States

The flowchart in Figure 20 defines the performance time allocation procedure in greater detail. The procedure begins by initializing the user performance time to zero and identifying the initial ancillary event for the allocation process. This event usually coincides with the primary event that marks the beginning of the specified performance period. In exceptional cases, it is the latest ancillary event prior to the start of that period. The start time for the first responsibility interval in the performance period is defined as the start of the period, and the overall responsibility state of the interval shall be determined in accordance with the specifications presented earlier.

The allocation procedure then identifies the successive ancillary events in the specified performance period. For each such event, the procedure determines the overall responsibility state subsequent to the event, and allocates performance time for the responsibility interval prior to the event. If the overall state prior to the event is "user responsible," the associated interval is added to the cumulative user performance time. If the prior state is "system responsible," the interval is ignored; i.e., system performance time is not recorded. If the overall state prior to the event is "split," the allocation of the associated interval depends on the responsibility state *subsequent* to the event in accordance with the principles outlined previously. Thus, if the subsequent state is "user responsible," the interval of "split" responsibility prior to the event is counted as user performance time. If the subsequent state is "system responsible," the "split" interval is regarded as system performance time. If the subsequent overall responsibility state is also "split," the intervals of "split" responsibility before and after the event are combined, and allocation of the aggregate interval is deferred until a subsequent interval of user or system responsibility is observed.

When the performance time allocation procedure observes an ancillary event that is coincident with or later than the end of the specified performance period, the end of the last responsibility interval is defined as the end of the performance period. If the last interval is one of overall user responsibility, that interval is added to the cumulative user performance time.

5. Data Analysis

This section outlines methods of analyzing measured performance data and defines statistical information that should be reported with measurement results. The section is divided into four subsections. The first three subsections outline basic data analysis methods and statistical reporting requirements corresponding, respectively, to the three general types of performance measurement experiments described in Section 2: absolute performance characterization, hypothesis testing, and analysis of factor effects. The fourth subsection describes more detailed analysis methods and associated reporting requirements for each experiment type.

Particular data analysis methods are described here only to the extent necessary to define the minimum requirements for reporting measurement precision. The subject of statistical data analysis is addressed comprehensively in other literature (e.g., see [4] and [8] through [11]). References [4] and [10] are of particular interest in that they describe available computer programs that implement widely used statistical data analysis techniques.

5.1 Absolute Performance Characterization. Absolute performance characterization experiments are undertaken to characterize the performance of a data communication service under a single specified set of conditions (a particular factor combination) without reference to factor effects or previously stated performance values. Although such experiments do not lead to decisions based on performance comparison, they are intended to be used in estimating population parameters from sample data. A parameter estimate derived from measurement cannot be expected to exactly equal the true population value because of sampling error, and it is therefore important that any such estimate be accompanied by an explicit specification of measurement precision. The primary purpose of the data analysis in absolute performance characterization experiments is to develop such a specification.

The precision of a population parameter estimate derived from a finite sample is expressed in terms of a confidence interval and an associated confidence level. A **confidence interval** is a range of values about

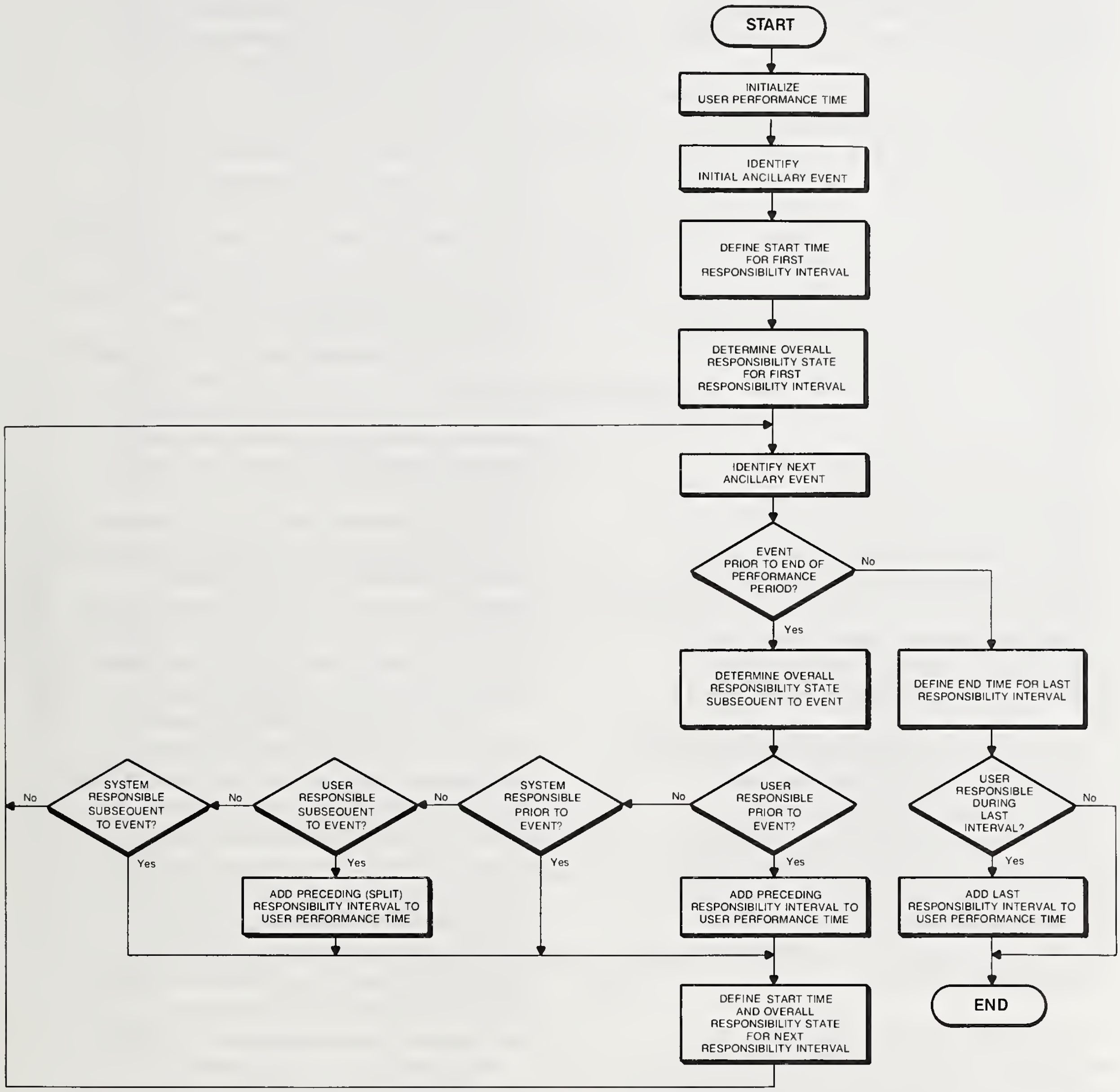


Figure 20
Performance Time Allocation Procedure

a measured parameter estimate within which the "true" (population) value of the parameter can, with a stated percent confidence, be expected to lie. The end points of a confidence interval are called **confidence limits.** These may be expressed either in absolute terms (e.g., ±0.1 second) or in relative terms (half-length of confidence interval divided by the estimate).

As defined in Section 2, a **confidence level** is a numerical value, typically expressed as a percentage, which describes the likelihood that a confidence interval calculated from sample data will contain the true value of the estimated parameter. If, for example, a 95% confidence level is specified, confidence intervals calculated from individual samples contain the "true" parameter value in about 95 out of 100 cases. Figure 21 illustrates a hypothetical set of 20 calculated 95% confidence intervals, one of which does not cover the true parameter mean.

Methods of calculating 90% and 95% confidence intervals for measured performance parameters are presented in [4] and implemented in the associated computer program. These or equivalent methods should be used in calculating confidence intervals for all absolute performance characterization experiments conducted in accordance with this standard. Confidence limits should always be reported with experiment results.

An experiment undertaken to characterize a single population may in fact reveal inhomogeneities among distinct population subsets. The significance of such differences can be tested using methods specified in [4]. Where significant differences are confirmed, the individual subpopulations should be characterized with separate performance values and confidence intervals.

5.2 Hypothesis Testing. This subsection outlines data analysis methods for hypothesis test experiments of the simplest kind, in which a performance value measured under a single factor combination is compared with a previously specified (hypothetical) value to determine whether a "significant" difference exists. A **statistical hypothesis** is an assumption about the distribution of a population, expressed in terms of specified values for one or more population parameters. A hypothesis test experiment is an experiment in which the validity of a particular statistical hypothesis is examined and ultimately accepted or rejected.[10] Such decisions are normally made with some uncertainty, since a parameter estimate based on a finite sample can deviate substantially from the true parameter value. The uncertainty of a hypothesis test experiment is expressed by its **significance level** α -- the probability of rejecting the tested hypothesis when in fact it is true.[11]

The hypothesis to be tested and a desired significance level are specified during test design. The data extraction and data reduction processes produce a measured estimate of the parameter in question. Given these inputs, an analysis to test the null hypothesis that a specified value equals the true population mean can be readily accomplished as follows:

(1) Calculate a confidence interval from the measured data using [4] or the associated computer program. If the null hypothesis is true, the calculated confidence interval includes the specified value with probability $(1-\alpha)$.

(2) Compare the specified value with the confidence interval. If the specified value lies within the confidence interval, the null hypothesis can be accepted with a significance level (probability of error) equal to α. If the hypothetical value lies outside the confidence interval, the null hypothesis is rejected.

In many hypothesis test experiments, the purpose is to determine whether actual performance is equal to *or better than* (rather than exactly equal to) a specified value. This approach can be applied in such experiments by simply halving the significance level. The resulting value then expresses the probability that an observed value lies on the "high performance" side of the confidence interval. Essentially the same approach can be used to test a negative hypothesis (that actual performance is worse than a specified value).

For comparability, this method or a compatible method should be used in analyzing

---

[10] The tested hypothesis is traditionally called the **null hypothesis**, because its truth implies that no difference exists between the hypothetical and true population values.

[11] Such an error is called a **type I error**. In a typical acceptance test, the type I error probability expresses the producer's (or service provider's) risk.

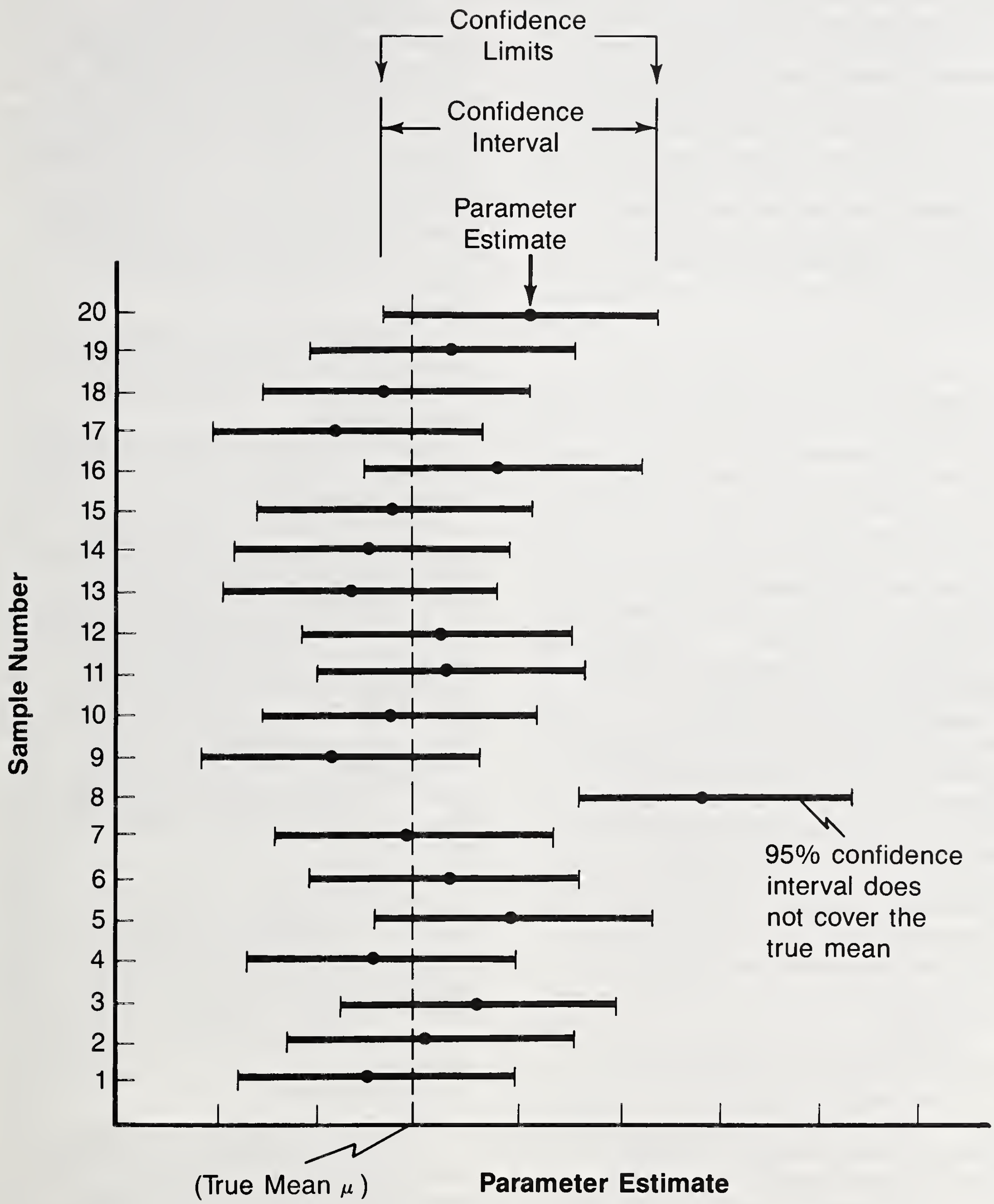


Figure 21
Illustration of Confidence Intervals

simple hypothesis test experiments conducted in accordance with this standard. A numerical significance level should always be reported with the results of such experiments.

In some hypothesis test experiments, it may be necessary to consider another type of error -- the error of accepting a stated hypothesis when it is actually false.[12] The likelihood of such an error is determined by three variables: the significance level (α) of the experiment, the test sample size, and the actual difference between the hypothetical and "true" values. Specific relationships among these variables may be determined from "power curves" as described in [8].

5.3 Analysis of Factor Effects. The final type of performance measurement experiment to be considered is the analysis of factor effects. In such experiments, tests are run under several factor combinations; and the results are compared in the analysis to identify and quantify postulated factor effects. The analysis typically consists of two steps:

(1) An analysis of variance (ANOVA) or equivalent categorical data analysis to identify the significant factor effects

(2) Individual performance comparisons to examine and quantify such effects

Analysis of variance is a statistical technique in which the observed variance of a sample is separated into several components, each describing the variability attributable to a particular factor. The variance attributed to each factor is compared with a residual variance, attributed to experimental error, and the factor effect is deemed significant (at a particular significance level, α) if the variances differ more than predicted by a calculated test statistic (the F statistic). The procedure is described in [8].

Analysis of variance should be used in evaluating the effects of performance factors on all of the time and rate parameters specified in ANSI X3.102-1983. An equivalent categorical data analysis using a chi-squared (χ^2) statistic should be used in the case of the failure probabilities. This analysis is described in [8]. Formulas for calculating the χ^2 and F statistics are presented in [4] and implemented in the associated computer program.

When an analysis of variance (or an equivalent categorical data analysis) indicates that a postulated factor has no significant effect on performance, data taken under all levels of the factor may be combined. This simplifies the overall performance specification by eliminating unnecessary categorization of the data. When significant factor effects are identified, individual performance comparisons are normally undertaken to examine those effects. Such comparisons may serve two objectives:

(1) They may simplify the specification as described previously by identifying *particular levels* of a performance factor that need not be distinguished.

(2) They may provide a basis for defining quantitative relationships between factor levels and performance values.

Performance data from different factor levels may be combined whenever one measured value lies within the confidence interval of another. The most direct way of summarizing quantitative relationships between factor levels and performance values is to simply list the calculated values (sample means) and confidence limits for each level. These data may also be graphed in various ways to suggest possible models of relationship [11]. More formal mathematical methods of expressing factor effects are described in the following subsection.

5.4 Additional Methods. This subsection describes three additional data analysis and presentation methods that may be used to provide more detailed information about a measured population. They are:

(1) Graphical presentation of frequency distributions

(2) Control charts

(3) Regression analysis

These three methods apply, respectively, to the three general types of performance measurement experiments defined earlier.

5.4.1 Graphical Presentation of Frequency Distributions. The relative frequency of occurrence of the possible outcomes of an experiment can be presented graphically in the form of histograms and cumulative distribution diagrams. A histogram is a graph that depicts

---

[12] Such an error is called a **type II error**, and its probability is commonly represented by the symbol β. In a typical acceptance test, the type II error probability expresses the consumer's (or user's) risk.

the relative frequencies of observed values within adjacent ranges or groups. In a typical histogram, the abscissa axis is divided into ranges of equal width, and the ordinate corresponding to each range represents the proportion of observed values falling within that range (Figure 22(a)). Histograms are useful in depicting how measured data are distributed -- in particular, the most frequently observed values.

A histogram is a sample result that may or may not be representative of the population from which the sample was taken. The precision with which the ordinates of a histogram represent the overall population distribution should be explicitly stated if inferences to the population are to be drawn. This can be done by placing confidence limits above and below each ordinate (Figure 22(b)). Confidence limits for a particular ordinate may be calculated from the total sample size (n) and the number of sample values falling within the corresponding range using [4] or the associated computer program. Such confidence limits define a range of values about the sample ordinate within which the true ordinate value can, with a stated percent confidence, be expected to lie. Note that these confidence limits apply to the individual ordinates rather than to the histogram as a whole.

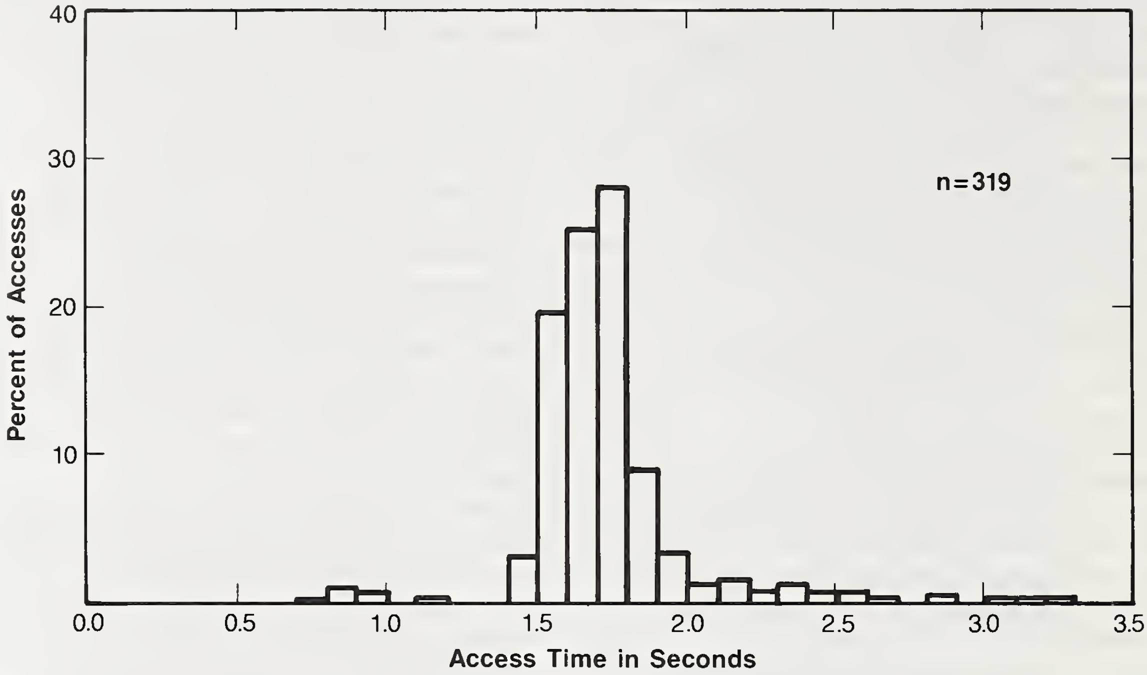
The **cumulative distribution function** (CDF) of a sample is the proportion of sample values less than or equal to any given value. A cumulative distribution diagram is a graphical presentation of a cumulative distribution function, with the possible observed values represented on the abscissa axis (Figure 23(a)). The "steps" in a cumulative distribution diagram normally represent individual observed values rather than groups of values. Cumulative distribution diagrams are useful in identifying the percentiles of a distribution (e.g., delay value exceeded on only 5% of the trials).

As in the case of a sample histogram, it is necessary to specify the precision of a sample cumulative distribution function if inferences to a population are to be drawn. This can be simply done by constructing a confidence band about the sample CDF (Figure 23(b)). The upper limit of the confidence band is calculated by adding a constant, determined from the sample size and the desired confidence level, to each CDF ordinate. The lower limit is calculated by subtracting the same constant from each CDF ordinate. The procedure and a table defining the appropriate constants are provided in [8]. If the stated confidence level is $1-\alpha$, confidence bands so constructed will completely contain the population cumulative distribution curve in $100(1-\alpha)\%$ of the cases.

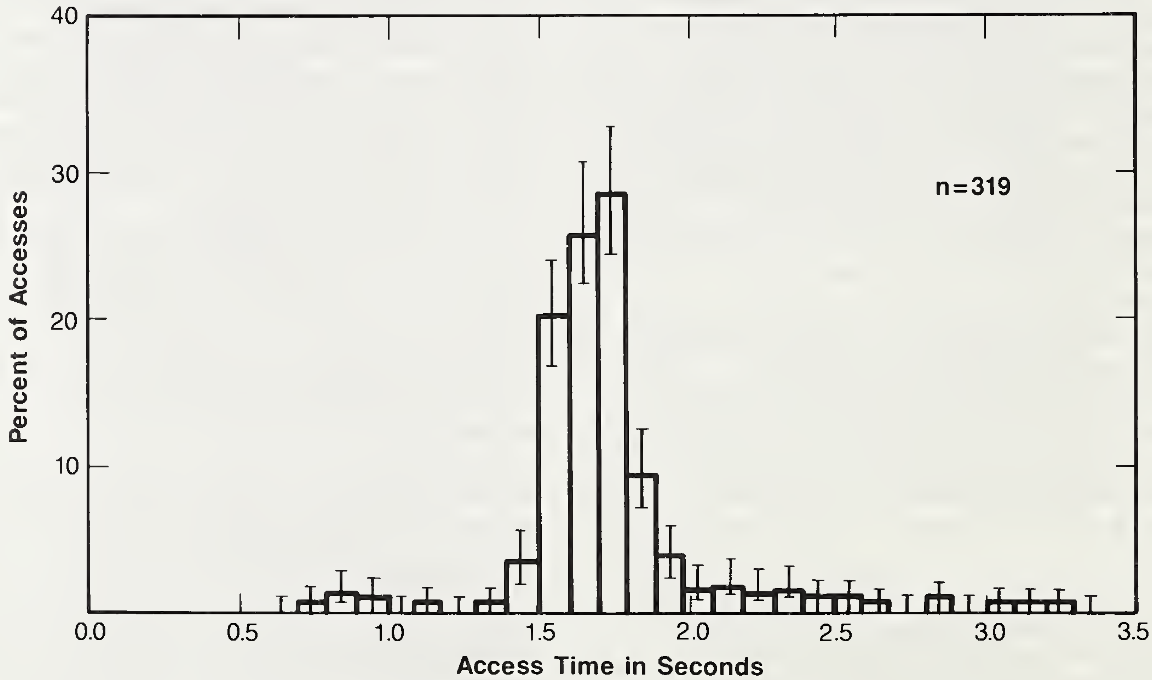
5.4.2 Control Charts. A control chart is a graphical means of detecting systematic (nonrandom) variation in a monitored process--in this application, the performance of a data communication service as measured by selected parameters. In a typical control chart, successive samples are numbered on the abscissa axis, and the ordinates represent the means of corresponding sampled values (Figure 24). The specified (or assumed) "true" value of the monitored parameter is indicated by a central line, and the allowable range of performance variability is delimited by upper and lower control limits. Each successive sample is then a test of the null hypothesis that the system is still "in control", i.e., that the true value of the monitored performance parameter has not changed. That hypothesis is rejected if any measured value falls outside the control limits. Control charts can be useful in detecting degradations in service quality (measured performance outside control interval on the low-performance side) as well as opportunities to improve network efficiency (measured performance outside control interval on the high-performance side).

The precision of a quality-control process is determined by the placement of the upper and lower control limits. These limits are usually set up symmetrically at $\pm \sigma 3/\sqrt{n}$ from the central line, where σ is the standard deviation of a single measurement of the controlled parameter and n is the sample size used in each performance measurement. For a normally distributed parameter, this placement of control limits ensures that the probability of falsely declaring a system out of control (i.e., the significance level, α) is less than 0.3%. Alternative methods of determining control limits may be used, but should be explicitly stated with any reported results. Specific procedures and tables for constructing control charts are provided in [8].

5.4.3 Regression Analysis. Regression analysis is a mathematical method of expressing relationships among random variables -- in this application, performance factors and

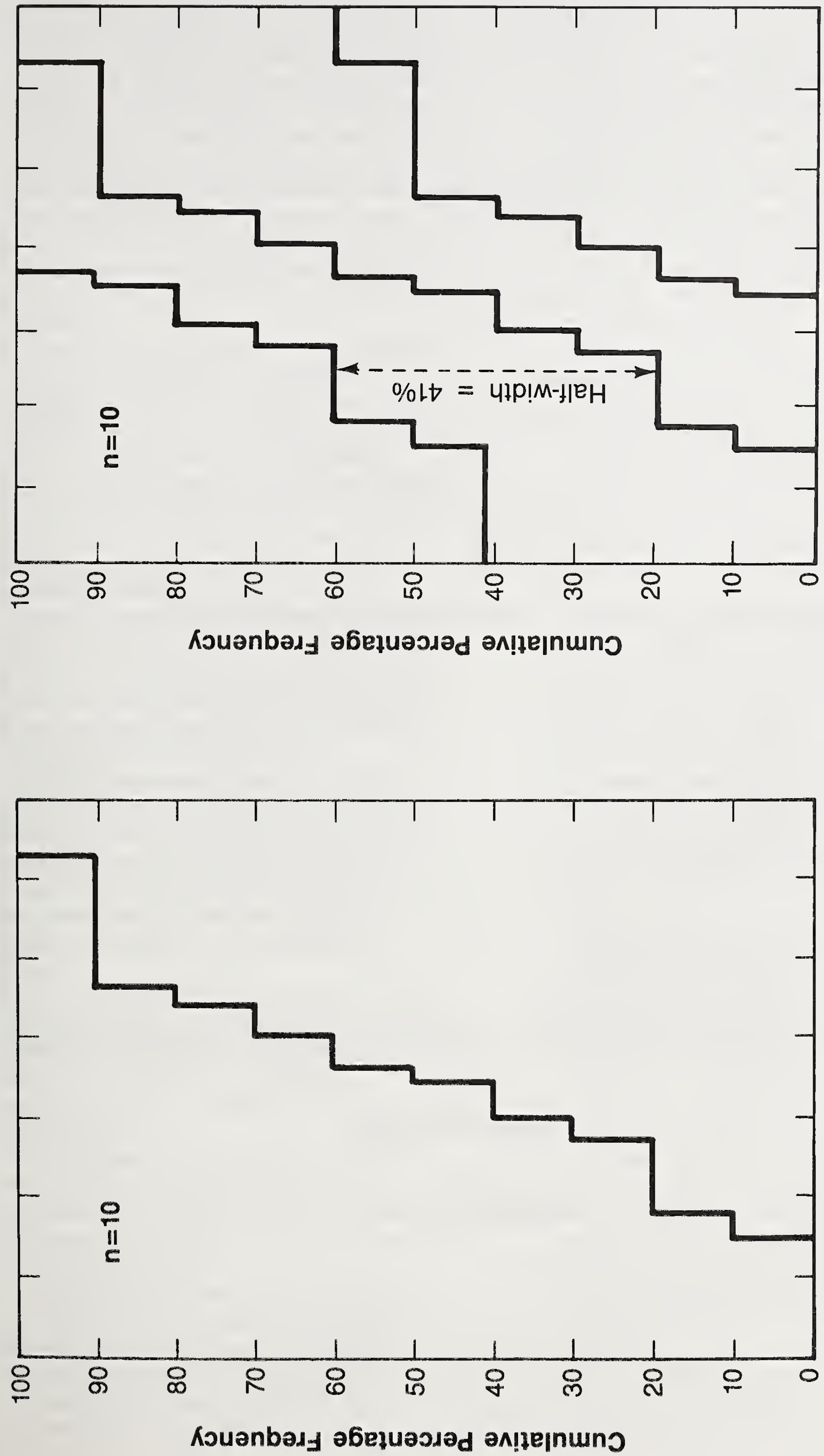


(*a*) *Typical Histogram*



(*b*) *The Same Histogram with 90% Confidence Limits*

Figure 22
Typical Histogram and Associated Confidence Limits



(a) A Cumulative Distribution Diagram

(b) The Same Diagram with Associated 95% Confidence Band

Figure 23
Typical Cumulative Distribution Diagram

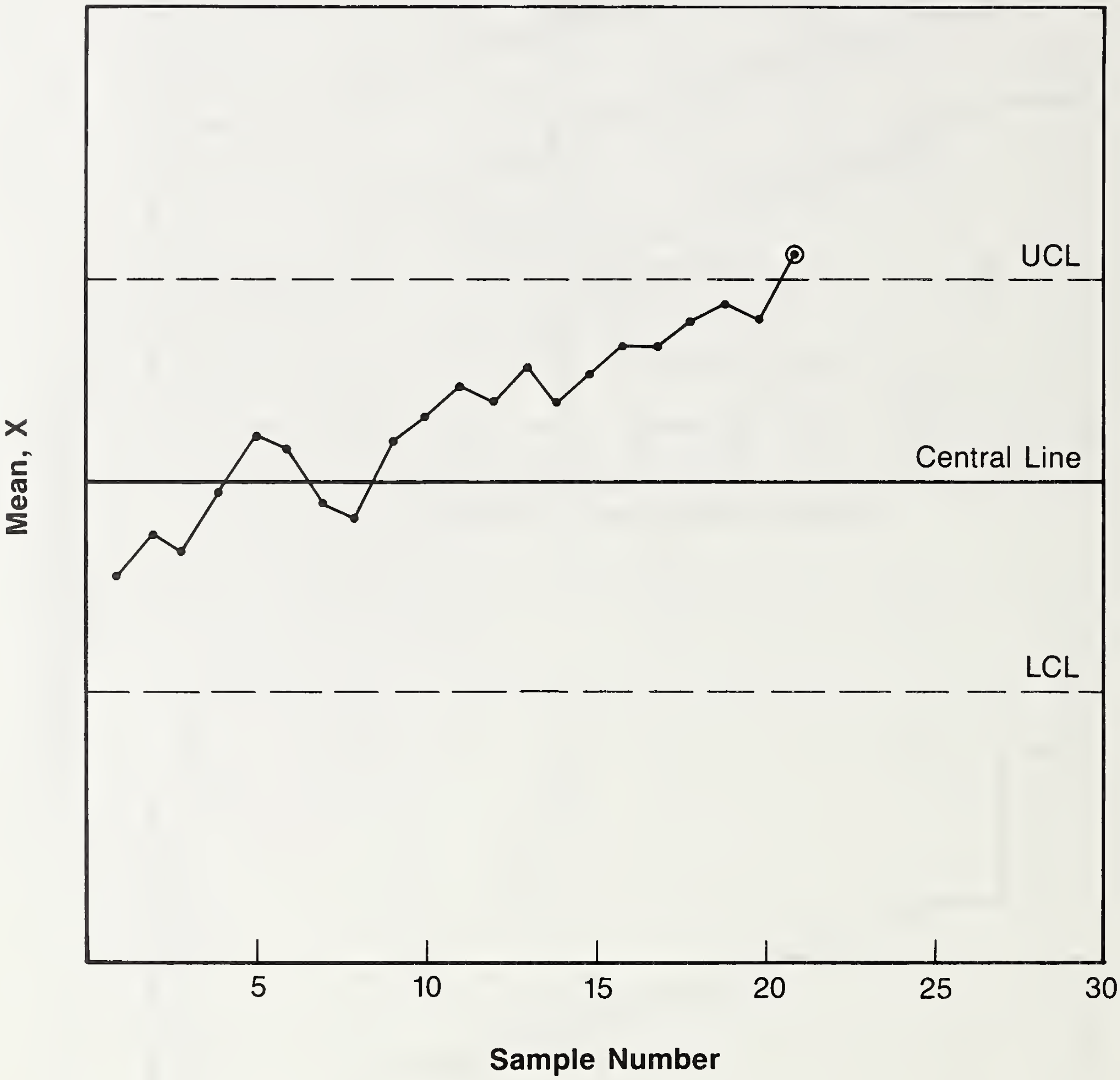


Figure 24
Typical Control Chart

parameters. In a typical regression analysis, a random sample of values for one variable (the "dependent" variable, y) is observed for each of several selected values of a second variable (the "independent" variable, x). The means of the various y distributions are then used in calculating a mathematical function, $y = f(x)$, which can be graphed (along with the means or the individual observed values) to illustrate the assumed relationship. The method most commonly used in fitting the assumed function (or curve) to the data is the method of least squares [9].

Figure 25(a) illustrates an important special case of regression analysis, called linear regression, in which the assumed function has the simple form $y = a + bx$. The slope b of such a "regression line" is called the **regression coefficient**. Examples of data communication performance relationships for which linear models have been proposed are transit delay as a function of hop distance (the number of communication link/switching node pairs separating source and destination users) and block error ratio as a function of block length (the size of a block in bits or time units). An important generalization of regression analysis is multiple regression, where several independent variables are considered.

A regression curve derived from sample data normally differ from the "true" population regression curve as a result of sampling error. If inferences are to be drawn from a sample regression curve, its precision in estimating the population regression curve shall be quantitatively stated. This can be done as follows:

(1) Calculate and report the "standard error of estimate" as described in [8]. This quantity represents the *y* variability about the regression curve; i.e., that part of the total *y* variability that is not accounted for by the regression curve.

(2) Calculate (and plot) confidence limits for representative ordinates of the sample regression curve (Figure 25(b)). Procedures and formulas for calculating such confidence limits are given in [8].

(3) Test the calculated regression coefficients (in the case of linear regression, the slope of the regression line) for significant difference from zero (as described in [8]). Results should include the specified significance level.

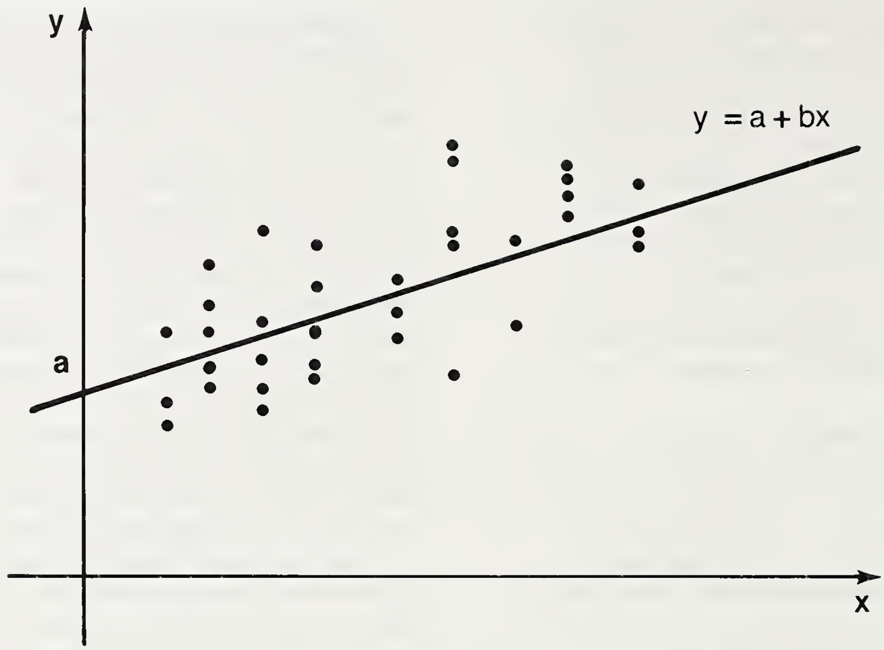
Regression analysis can also be applied in analyzing the results of experiments in which each of the variables of interest is sampled randomly; i.e., none of the variables is controlled by the experimenter. Such a study is often called a **correlation analysis.** (An equivalent and sometimes preferable analysis technique in the transform domain is **spectral analysis.**) The linear regression examples cited earlier could be cited as correlation examples if the previously "independent" variables (e.g., hop distance, block length) were allowed to vary randomly during the experiment. Correlation analysis is appropriate in examining relationships between performance parameter pairs (e.g., throughput and delay) where neither is under experimental control.

The most common graphical representation of correlation analysis results is the scatter diagram (Figure 26(a)). The data plotted in such a diagram are randomly observed pairs of values (x, y) for two performance variables of interest.

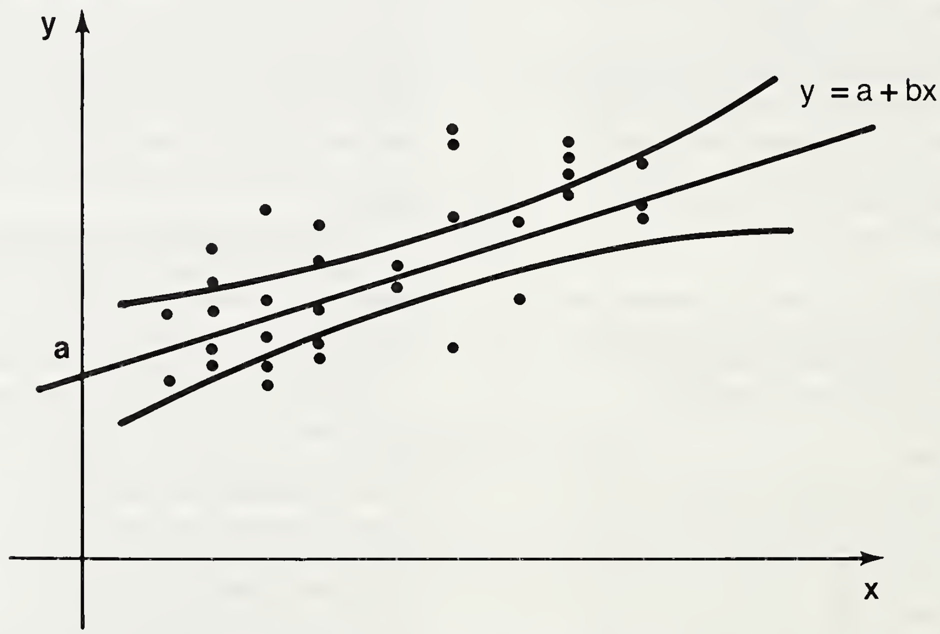
It is often useful to fit a line or curve to a plotted scatter diagram in order to predict one variable from the other. This can be done using the regression methods described earlier, but two regression functions may now be defined: the regression of *y* on *x* and the regression of *x* on *y* (Figure 26(b)). Normally, the variable to be predicted is labeled *y* and only the former function is plotted. Careful judgement is required in selecting the dependent and independent variables, and in curve fitting.

The **sample correlation coefficient** is a dimensionless number, in the range $-1 \leq r \leq 1$, which expresses the degree of linear dependence between observed values for two (or more) random variables [9]. The value of r is ± 1 if, and only if, one variable completely determines the other; i.e., all observed points lie on the regression line. The sign of the sample correlation coefficient is positive if the slope of the regression line is positive and vice versa. If $r = 0$, the sample values are uncorrelated and the regression line is horizontal. The sample correlation coefficient can be useful in comparing widely different sets of data because of its dimensionless property.

The precision with which a regression curve fitted to the sample data in a scatter diagram approximates the population regression curve

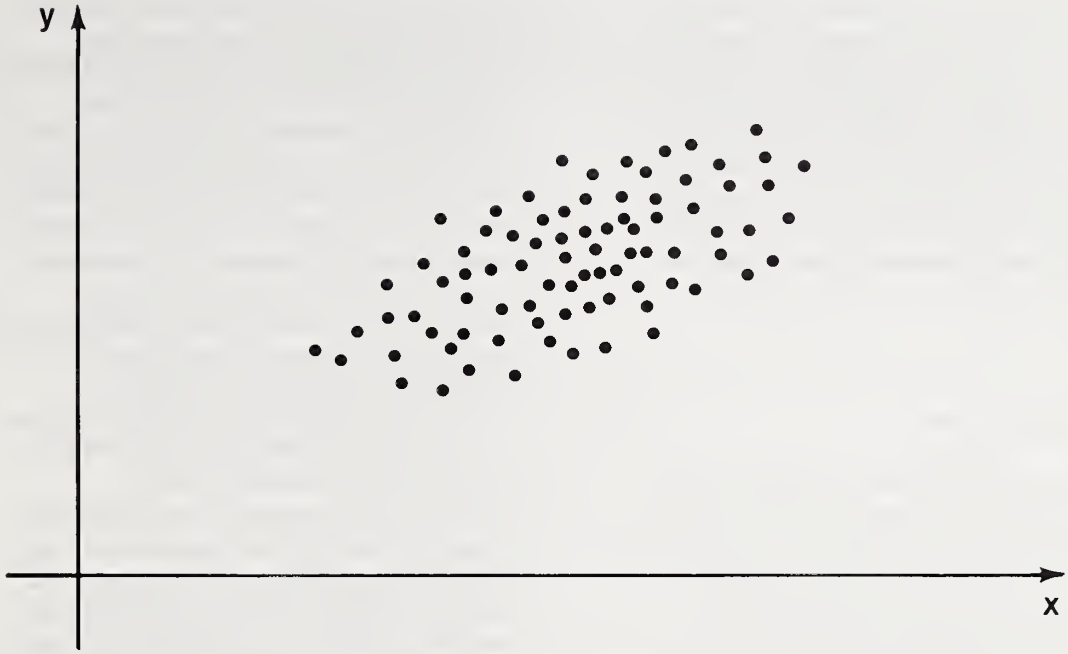


(a) Regression Curve

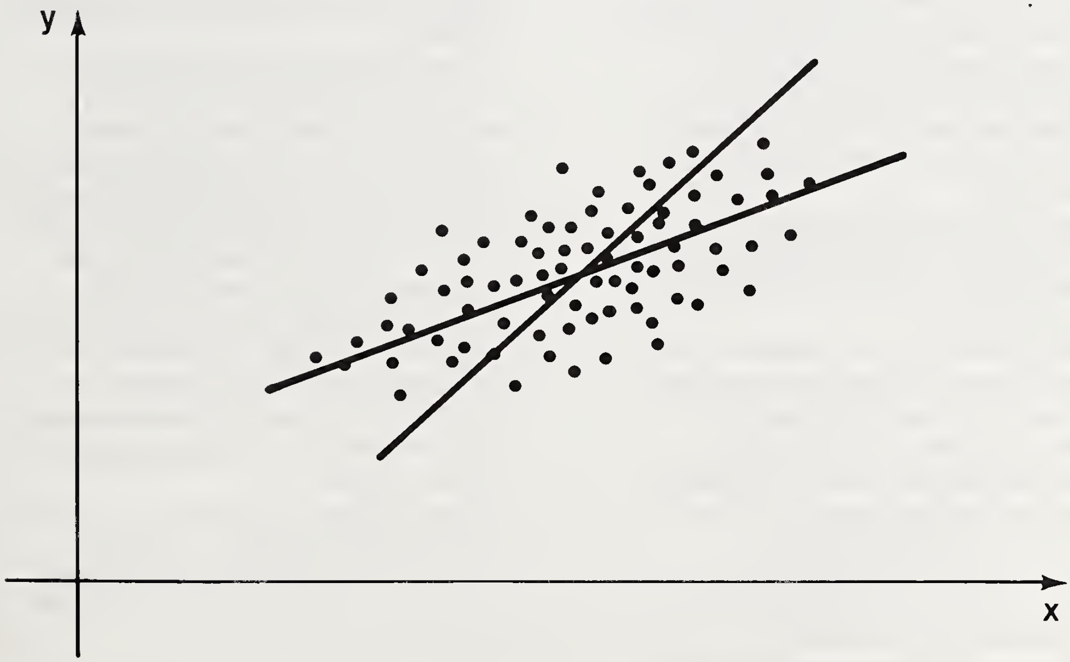


(b) The Same Curve with Confidence Limits

Figure 25
Typical Regression Curve and Associated Confidence Limits



(a) Scatter Diagram



(b) The Same Diagram with the Two Associated Regression Lines

Figure 26
Scatter Diagram and Associated Regression Lines

can be expressed by the same three measures described earlier: (1) the standard error of estimate, (2) confidence limits for the regression curve, and (3) a significance test of the sample regression coefficient. A significance test can be conducted on the sample correlation coefficient as an alternative to the latter. A high correlation between variables does not necessarily imply cause and effect, although it may indicate support for such a conclusion.

6. Summary Forms

This section provides example forms that may be used in summarizing performance measurement experiments conducted in accordance with this standard. The forms are of two types: those that summarize the experiment design, and those that summarize the measurement results. Their use is optional. Examples of completed forms are provided in Appendix B.

6.1 Experiment Design Summary Forms. A form of the type shown in Table 9 can be used to summarize performance measurement experiment designs. Normally, completion of such a form also summarizes the data collection process, since that process is governed by the experiment design. Where the data collection process actually used in an experiment differs from that planned in the design the former should be specified, with significant differences noted.

Table 9 is comprised of seven parts corresponding to the 7-step experiment design procedure defined in Section 2. It shall be completed as follows:

(1) Briefly summarize the decision context of the experiment to indicate how the results are to be used (e.g., acceptance testing, optimization of routing). Define the experiment type (absolute performance characterization, hypothesis testing, or analysis of factor effects). In the case of hypothesis test experiments, state the tested hypothesis.

(2) List or otherwise identify the performance parameters that were measured.

(3) Characterize the user pairs of interest in the experiment and identify the overall set of user pairs represented. Define the observation period of the overall experiment and the times selected for individual tests. Define the user/system interfaces monitored (e.g., DTE X.25 Level 3). Define event sequences characterizing the data communication sessions observed, by reference to either (a) an attached data communication session profile, or (b) a specified user/system interface protocol such as X.25. Identify the reference event or events corresponding to each defined interface event. Identify the session type (connection-oriented or connectionless). Specify the performance timeouts and thresholds used (e.g., maximum access time, Transfer Denial threshold for Bit Error Probability).

(4) List all performance factors considered in the experiment. For each factor, identify the specific level or levels tested.[13]

(5) Identify the method used in selecting the performance trials to be measured from the defined population, including any departures from randomization such as stratified sampling. Define the sample sizes obtained and the expected confidence (or significance) levels.

(6) List the individual tests conducted during the experiment, and identify the factor combination used in each test.

(7) Where a mathematical model is developed to summarize the experiment design, present the model equations and define each variable.

It will often be useful to augment the information in Table 9 with one or more attachments. Tables 10 and 11 provide standard forms for two such attachments. Table 10 defines a framework for the graphical presentation of data communication session profiles. Symbols commonly used in such profiles are defined in the legend. The primary and ancillary reference event numbers indicated in the legend are defined in Tables 3 and 5, respectively. Where necessary, an individual data communication session profile may be divided between several consecutively numbered sheets.

Table 11 lists the 14 performance timeouts and thresholds that shall be reported in summarizing a comprehensive performance

---

[13] Parameter values are not compared among factor combinations in absolute performance characterization and simple hypothesis test experiments, but the relevant factors and tested levels should still be stated to ensure proper interpretation of the measurement results.

Table 9
Experiment Design Summary Form

EXPERIMENT DESIGN SUMMARY

1. Experiment Objectives ______________________

2. Measured Parameters ______________________

3. Population Definition
   a. User Pair Characteristics ______________________
   b. User Pairs Represented ______________________
   c. Observation Period(s) ______________________
   d. User/System Interface Characteristics ______________________
   e. Session Profile(s)* ______________________
   f. Reference Events (and Session Type)* ______________________
   g. Timeouts/Thresholds* ______________________

4. Performance Factors

   Factor — Level(s)

5. Population Sample
   a. Method of Selection ______________________
   b. Sample Size ______________________
   c. Confidence (or Significance) Level ______________________

6. Test Conditions

   Test Number(s) — Factor Combinations

7. Mathematical Model (if used)

*May be specified in attached forms.

Table 10
Data Communication Session Profile Form

DATA COMMUNICATION SESSION PROFILE (Sheet ______ of ______)

User	Data Communication System	User

Legend

(Name) →	Interface Signal (User Input to System)
← (Name)	Interface Signal (System Output to User)
0/1 • • • 9/6	Reference Event Number (Primary/Ancillary)
— — — —	Cause/Effect Linkage Between Signals

Table 11
Timeout and Threshold Specification Form

PERFORMANCE TIMEOUT AND THRESHOLD SPECIFICATIONS

1. Performance Timeouts
   a. Access ____________
   b. User Information Block Transfer ____________
   c. Transfer Sample Input/Output ____________
   d. Source User Disengagement ____________
   e. Destination User Disengagement ____________

2. User Performance Time Fractions
   a. Access ____________
   b. User Information Block Transfer ____________
   c. Transfer Sample Input/Output ____________
   d. Source User Disengagement ____________
   e. Destination User Disengagement ____________

3. Transfer Denial Criteria
   a. Transfer Sample Size ____________
   b. Bit Error Probability Threshold ____________
   c. Bit Loss Probability Threshold ____________
   d. Extra Bit Probability Threshold ____________
   e. User Information Bit Transfer Rate Threshold ____________

measurement. They consist of 5 performance timeouts, 5 user performance time fractions, and 5 Transfer Denial criteria. These are defined and described in ANSI X3.102-1983. In experiments in which a subset of the performance parameters is measured, only the pertinent timeouts and thresholds need be specified.

6.2 Measurement Results Summary Forms. Tables 12 through 14 are forms that may be used in summarizing performance measurement results. A separate form is provided for each of the three experiment types defined earlier: absolute performance characterization, hypothesis testing, and analysis of factor effects.

Table 12 defines the basic results to be reported in summarizing absolute performance characterization experiments. These results include an estimated value (sample mean) for each measured parameter; the corresponding confidence levels; and the upper and lower confidence limits for each estimate as calculated from the measured data. Supplementary information such as histograms and distribution statistics may also be reported.

Table 13 defines information to be reported in summarizing simple hypothesis test experiments. This information includes, for each measured parameter, (1) the hypothetical value specified in the experiment design; (2) the criterion for accepting the tested hypothesis; (3) the estimated value and associated confidence limits calculated from the measured data; and (4) whether the tested hypothesis is "Accepted" or "Rejected" on the basis of the experiment results. As discussed in Section 5, the null hypothesis (that the specified value equals the true population mean) is accepted if the specified value lies within the calculated confidence interval; otherwise, that hypothesis is rejected. In "one-sided" hypothesis test experiments, where the tested hypothesis is that actual performance is equal to or better than a specified value, only the confidence limit on the "low performance" side of each estimated value need be reported.

Table 14 defines the information to be reported in summarizing an analysis of factor effects. A separate table should be completed for each tested factor. In each table, the following information is specified: (1) the performance factor in question; (2) the tested levels of that factor; (3) the estimated parameter values for each factor level; and (4) for each measured parameter, the significance level at which the estimated values differ. Where an analysis of factor effects reveals that differences between certain factor levels are not significant, the corresponding estimated values may be combined.

7. References

[1] American National Standard for Information Systems-Data Communication Systems and Services - User-Oriented Performance Parameters, ANSI X3.102-1983. Available from: American National Standards Institute, 1430 Broadway, New York, NY, 10018. When this American National Standard is superseded by a revision approved by the American National Standards Institute, Inc., the revision shall apply.

[2] Cox, D.R. Planning of experiments. New York: John Wiley and Sons, Inc., 1958.

[3] Wortendyke, D.R.; Seitz, N.B.; Spies, K.P.; Crow, E.L.; Grubb, D.S. User-oriented performance measurements on the ARPANET: The testing of a proposed Federal Standard. Report - National Telecommunications and Information Administration, 1982; Report No 82-112. Available from: National Technical Information Service, Springfield, VA, 22161.

[4] Miles, M.J. Sample size and precision in communication performance measurements. Report - National Telecommunications and Information Administration, 1984; Report No 84-153. Available from: National Technical Information Service, Springfield, VA, 22161.

[5] Bell System Technical Reference. 1969-70 switched telecommunications network connection survey. New York: American Telephone and Telegraph Company, 1971; Publication No 41007. Available from: AT&T Customer Information Center, P.O. Box 19901, Indianapolis, IN, 46219.

[6] Kamas, G.; Howe, S.L. Time and frequency user's manual. Washington, DC: National Bureau of Standards, November 1979; National Bureau of Standards Special Publication No

559. Available from: Superintendent of Documents, U.S. Government Printing Office, Washington, DC, 20402.

[7] Seitz, N.B.; Grubb, D.S. American National Standard X3.102 user reference manual. Report - National Telecommunications and Information Administration, 1983; Report No 83-125. Available from: National Technical Information Service, Springfield, VA, 22161.

[8] Crow, E.L.; Davis, F.A.; Maxfield, M.W. Statistics manual. New York: Dover Publications, Inc., 1960.

[9] Dixon, W.J.; Massey, F.J. Introduction to statistical analysis, 3rd ed. New York: McGraw-Hill, Inc., 1969.

[10] Hogben, D.; Peavy, S.T.; Varner, R.N. OMNITAB II user's reference manual. National Bureau of Standards, 1971; Technical Note No 552. Available from: Superintendent of Documents, U.S. Government Printing Office, Washington, DC, 20402.

[11] Tukey, J.W. Exploratory data analysis. Reading, MA: Addison-Wesley Publishing Company, 1977.

Table 12
Measurement Results Summary
(Absolute Performance Characterization Experiment)

PERFORMANCE PARAMETER	ESTIMATED VALUE	ESTIMATE PRECISION		
		CONFIDENCE LEVEL	LOWER CONFIDENCE LIMIT	UPPER CONFIDENCE LIMIT
1. Access Time				
2. Incorrect Access Probability				
3. Access Denial Probability				
4. Access Outage Probability				
5. Bit Error Probability				
6. Bit Misdelivery Probability				
7. Bit Loss Probability				
8. Extra Bit Probability				
9. Block Transfer Time				
10. Block Error Probability				
11. Block Misdelivery Probability				
12. Block Loss Probability				
13. Extra Block Probability				
14. User Information Bit Transfer Rate				
15. Transfer Denial Probability				
16. Disengagement Time				
17. Disengagement Denial Probability				
18. User Fraction of Access Time				
19. User Fraction of Block Transfer Time				
20. User Fraction of Sample Input/Output Time				
21. User Fraction of Disengagement Time				

Table 13
Measurement Results Summary
(Hypothesis Test Experiment)

TESTED HYPOTHESIS:
SIGNIFICANCE LEVEL:

PERFORMANCE PARAMETER	SPECIFIED VALUE	DECISION CRITERION	MEASUREMENT RESULTS			
			ESTIMATED VALUES			DECISION
			LOWER CONFIDENCE LIMIT	MEAN	UPPER CONFIDENCE LIMIT	
1. Access Time						
2. Incorrect Access Probability						
3. Access Denial Probability						
4. Access Outage Probability						
5. Bit Error Probability						
6. Bit Misdelivery Probability						
7. Bit Loss Probability						
8. Extra Bit Probability						
9. Block Transfer Time						
10. Block Error Probability						
11. Block Misdelivery Probability						
12. Block Loss Probability						
13. Extra Block Probability						
14. User Information Bit Transfer Rate						
15. Transfer Denial Probability						
16. Disengagement Time						
17. Disengagement Denial Probability						
18. User Fraction of Access Time						
19. User Fraction of Block Transfer Time						
20. User Fraction of Sample Input/Output Time						
21. User Fraction of Disengagement Time						

Table 14
Measurement Results Summary
(Analysis of Factor Effects)

PERFORMANCE PARAMETER	PERFORMANCE FACTOR:					
	ESTIMATED VALUES					SIGNIFICANCE LEVEL
	FACTOR LEVEL					
	1	2	3	4	5	
1. Access Time						
2. Incorrect Access Probability						
3. Access Denial Probability						
4. Access Outage Probability						
5. Bit Error Probability						
6. Bit Misdelivery Probability						
7. Bit Loss Probability						
8. Extra Bit Probability						
9. Block Transfer Time						
10. Block Error Probability						
11. Block Misdelivery Probability						
12. Block Loss Probability						
13. Extra Block Probability						
14. User Information Bit Transfer Rate						
15. Transfer Denial Probability						
16. Disengagement Time						
17. Disengagement Denial Probability						
18. User Fraction of Access Time						
19. User Fraction of Block Transfer Time						
20. User Fraction of Sample Input/Output Time						
21. User Fraction of Disengagement Time						

Appendixes (These Appendixes are not part of American National Standard X3.141-1987, but are included for information only.)

Appendix A

Glossary of Terms

This Appendix provides definitions for specialized data communication performance assessment terms used in the standard. Definitions provided in a companion standard, ANSI X3.102-1983, are repeated here for convenience. Definitions directly adopted from other standards are identified and referenced.

access request. The event that notifies the system of a user's desire to initiate a data communication session. It begins the access function and starts the counting of access time. Two specific examples of Access Request events are the off-hook event in the public switched telephone network, and the completion of a Connect request by a terminal operator in the ARPANET.

accuracy. A general performance criterion expressing the correctness with which a specific communication function is accomplished.

aggregate user (of a subsystem). Any collection of entities outside a defined subsystem, comprising one or more end users and the data communication system elements that connect those users with the subsystem.

blocking signal. A signal or communication issued by a user or the data communication system to indicate inability to complete a communication function in progress. Such signals are overhead information. Examples of blocking signals are circuit busy and user busy signals.

committed state. A user condition, relative to a particular data communication session, in which (1) the user has agreed to participate in the session and (2) the user intends to transmit or receive user information. Changes in user commitment are normally a result of user-initiated interface signals (e.g., Access and Disengagement requests).

confidence interval. A range of values about a measured parameter estimate within which the "true" (population) value of the parameter can, with a stated probability, be expected to lie.

confidence level. A numerical value, typically expressed as a percentage, which describes the likelihood that a confidence interval calculated from sample data will contain the true value of the estimated parameter. The corresponding specification in hypothesis testing is the significance level, which can be expressed as the complement of the confidence level.

confidence limits. A statistical term denoting the end points of a confidence interval. Confidence limits provide a means of stating the precision of parameter estimates based on a finite sample size.

correlation analysis. An analysis of experimental data in which each of the variables of interest is sampled randomly; i.e., none of the variables is controlled by the experimenter.

cumulative distribution diagram. A graphical presentation of a cumulative distribution function, with the possible observed values represented on the abscissa axis. (See Figure 21(a) in this standard.)

cumulative distribution function (CDF). A nondecreasing function of a random variable giving the proportion of sample values less than or equal to any given value of the variable.

data communication service. A specified user information transfer capability provided by a data communication system to two or more end users.

data communication session. A coordinated sequence of user and system activities whose purpose is to cause digital user information present at one or more source users to be transported and delivered to one or more destination users. A normal data communication session between a user pair comprises: (1) an access function, (2) a user information transfer function, and (3) a disengagement function for each user. A data communication session is formally defined by a data communication session profile.

data communication session profile. The exact sequence of user/system interface signals by which data communication service is provided in a typical (successful) instance. A complete data communication session profile should also include any possible blocking (service refusal) sequences for a particular set of users.

data communication subsystem. A group of data communication system elements terminated within the end user interfaces.

data communication system. A collection of transmission facilities and associated switches, data terminals, and protocols that provide data communication service between two or more end users. The data communication system includes all functional and physical elements that participate in transferring information between end users. The system element that interfaces with the end user is a data terminal or a computer operating system. A computer operating system normally serves as the first point of contact for application programs requiring data communication service.

data communication user. Either (1) an end user of a data communication system, or (2) an aggregate user of a data communication subsystem.

data medium. Either (1) the material in which or on which a specific physical variable may represent data or (2) the physical quantity that may be varied to represent data. (See American National Dictionary for Information Processing, X3/TR-1-77.)

destination user. The user to whom a data communication system is to deliver a particular user information block (or unit).

destination user information bits. The binary representation of user information transferred from a data communication system to a destination user. When the user information is output as nonbinary symbols (e.g., alphanumeric characters), the destination user information bits are the bits on which a final decoding is performed to generate the delivered symbols.

disengagement confirmation. The event that verifies that a particular user's participation in an established data communication session has been terminated. For that user, it completes the disengagement function and stops the counting of disengagement time. Disengagement Confirmation occurs, for a particular user, when (1) disengagement of that user has been requested; and (2) the user is able to initiate a new access attempt. Most data communication systems notify the user that a new access attempt can be initiated by issuing an explicit Disengagement Confirmation signal. An example is the system's printing of a Closed message at an operator terminal in the ARPANET. In cases where no Disengagement Confirmation signal is issued, the user may initiate a new access attempt to confirm disengagement.

disengagement request. The event that notifies the system of a user's desire to terminate an established data communication session. It is complementary to the Access Request in most systems. Each Disengagement Request begins the disengagement function and starts the counting of disengagement time for one or both users. Disengagement is normally requested simultaneously for both users in connection-oriented sessions, and independently for each end user in connectionless sessions. Examples of Disengagement Requests are a user's hanging up in the public switched telephone network, and the terminal operator's typing of a Close request (after successful access) in the ARPANET.

end of block transfer. The event that completes the transmission of a specified user information block between the source and destination users. That event occurs, for any given user information block, when (1) all bits in the block are physically present within the destination user facility, and (2) the destination user has been notified that the information is available for use. The notification may be explicit or implicit. For each block, the End of Block Transfer event stops the counting of block transfer time. When the received block precedes the first block in a transfer sample, End of Block Transfer begins output of that sample and starts the counting of sample output time. When the received block is the last block in a transfer sample, End of Block Transfer completes the collection of the sample and stops the counting of sample output time. An example of an End of Block Transfer event is the completion of printing of a line of text at a destination terminal.

end user. Either (1) the human operator of a data terminal, with any associated data medium or (2) a computer application program that utilizes communicated information, with any associated data medium.

end user interface. See *user/system interface*.

factor. A system or usage variable identified in a particular experiment as influencing, or possibly influencing, measured values for the parameters described in ANSI X3.102-1983.

factor combination. A set specifying a particular level for each factor of interest in an experiment.

fractional factorial experiment. An experiment in which only a subset of the possible factor combinations is tested.

full factorial experiment. An experiment in which every possible combination of the defined factor levels is tested at least once.

functional interface. Any data communication system boundary. A functional interface may or may not correspond to a physical interface. One example of a user/system functional interface is the protocol or command boundary between an application program and the operating system in a remote-access data processing computer.

independent disengagement attempt. A disengagement attempt whose successful completion does not require a concurring response from the user not initiating the disengagement request.

interface event. Any discrete transfer of information across a user/system interface.

level. One of a defined set of states or values a given factor assumes in an experiment.

measured value. An estimate of the true (population) value of a parameter, obtained by measurement of a sample in one or more tests.

negotiated disengagement attempt. A disengagement attempt whose successful completion requires a concurring response from the user not initiating the disengagement request.

nonoriginating user. In a data communication session, the user that does not initiate the Access Request.

nonoriginating user commitment. The expression, via a communicated interface signal, of a nonoriginating user's willingness and ability to participate in a requested data communication session. That event occurs during access only in connection-oriented sessions. It is used in identifying Incorrect Access (e.g., misconnection) outcomes. The most familiar example is the called user's answering of a normal telephone call. Issuance of an OPEN ANY HOST (LISTEN) system call by an application program in the ARPANET is another.

null hypothesis. The statistical hypothesis tested in an experiment--so called because its truth implies that no difference exists between the hypothetical and true population values.

overhead information. All information other than user information. Overhead information includes (1) information transferred from a user to the system for the purpose of controlling internal system operations, (2) information transferred from the system to a user for the purpose of reporting system status or controlling user operations, and (3) information transferred between distinct elements of a system to control their joint operations, i.e., information neither input from nor output to a user. Examples are: ASCII DLE, ESC, and ENQ characters; circuit busy and ringing signals in the public switched network; and the flag, address, control, and FCS fields of ADCCP frames. (See American National Standard for Advanced Data Communication Control Procedures (ADCCP), ANSI X3.66-1979.)

performance criterion. A general category of user concern within which various related performance parameters may be grouped. Three performance criteria are defined in ANSI X3.102-1983: speed, accuracy, and reliability. These criteria do not themselves assume values; rather, they serve as a conceptual framework for organizing the parameters.

performance parameter. A statistical quantity whose numerical values characterize a particular aspect of data communication system performance. In this standard, a population mean.

population. A set comprising all trials of interest in a particular experiment. A population may be comprised of any one of three types of trials: access, user information transfer, or disengagement.

precision. The closeness of trial values to each other, as measured, for example, by their standard error. Distinguished from statistical accuracy, which describes the closeness of an estimate to the true population value.

random sample (of a statistical population). A subset of the population chosen in such a way that each member of the population has an equal chance of being included in the sample. Random samples may be selected with the aid of random number tables.

reference event. A generic event that subsumes many system-specific interface events having a common performance significance. The reference events collectively specify all information needed to describe performance in accordance with this standard and ANSI X3.102-1983.

regression coefficient. The slope b of a regression line, when the linear regression has the assumed form $y = a + bx$. (There can be several regression coefficients $b_1, b_2, \ldots, b_k$ in a multiple linear regression $y = a + b_1x_1 + b_2x_2 + \cdots + b_kx_k$.)

reliability. A general performance criterion expressing the probability that a specific data communication function will be performed successfully for a specified time period.

sample correlation coefficient. A dimensionless number, in the range $-1 \leq r \leq 1$, that expresses the degree of linear dependence between observed values of two random variables. The value of r is ± 1 if, and only if, one variable completely determines the other; i.e., all observed points lie on the regression line. The sign of the sample correlation coefficient is positive if the slope of the regression line is positive and vice versa. If $r = 0$, the sample values are uncorrelated and the regression line is horizontal. The sample correlation coefficient can be useful in comparing widely different sets of data because of its dimensionless property.

service time interval. The specific interval or intervals of time throughout the 24-hour day during which a data communication service supplier agrees to make a digital data communication service available to a particular user.

significance level. In a simple hypothesis test experiment, the probability of rejecting the tested hypothesis when in fact it is true.

source user. A user from whom a data communication system receives a particular user information block.

source user information bits. The binary representation of user information transferred from a source user to a data communication system for ultimate delivery to a destination user. When the user information is input in nonbinary form (e.g., keyboard entries), the source user information bits are the bits used to encode these symbols initially; any bits added to this initial encoding for purposes such as error control, flow control, and polling are overhead bits rather than source user information bits.

speed. A general performance criterion expressing the rapidity with which a specific data communication function is performed. Speed is described in terms of performance times and rates.

start of block input to system. The event that transfers one or more bits at the beginning of a user information block from the source user to the data communication system. That event usually coincides with the start of transfer of the block. However, the two events differ in cases where the system provides an input buffer (e.g., the one-line input buffer provided in many CRT terminals). In such cases, Start of Block Input is defined to occur when the first bit of user information is physically stored in that buffer. An example is the operator's typing of the first user information character at a buffered CRT terminal. When the input block is the first block in a data communication session (after Nonoriginating User Commitment in connection-oriented sessions), Start of Block Input completes the access function, stops the counting of access time, and identifies the start of the user information transfer function. The reason for using the start of input of user information (rather than a "connection open" or similar system response) to define the end of access is that such responses are not provided in all systems.

start of block transfer. The event that initiates actual transmission of a specified user information block between the source and destination users. That event occurs, for any given user information block, when (1) all bits in the block are physically present within the system facility, and (2) the system has been authorized to transmit that information. Authorization may either be an explicit user action (e.g., typing Carriage Return at a buffered CRT terminal) or an implicit part of inputting the user information itself (e.g., typing a single character at an unbuffered asynchronous terminal). For each block, the Start of Block Transfer event begins the counting of block transfer time. When the transmitted block precedes the first block in a transfer sample, Start of Block Transfer begins collection of that sample and starts the counting of sample input time (a transfer sample includes the interblock gap preceding the first block in the sample). When the transmitted block is the last block in a transfer sample, Start of Block Transfer completes input of the sample and stops the counting of sample input time.

statistical hypothesis. An assumption about the distribution of a population, often expressed in terms of specified values for one or more population parameters.

subsystem. See *data communication subsystem.*

subsystem interface. The physical or functional boundary delimiting a subsystem. Each subsystem interface also identifies a collection of entities outside the subsystem, comprising one or more end users and the data communication system elements that connect those users with the subsystem, called an *aggregate user.*

supported performance parameter. One of four user information transfer performance parameters used in the measurement of Transfer Denial Probability. The four are: Bit Error Probability, Bit Loss Probability, Extra Bit Probability, and User Information Bit Transfer Rate.

system. See *data communication system.*

system blocking signal. The event that informs a user that the system cannot provide communication service on a particular request because some required system facility is unavailable. The required facility (e.g., trunk circuit) may be unavailable because it is serving another user, or because it is in an outage condition; the two possibilities often cannot be distinguished at the end user interface. System Blocking Signals are used to identify Access Denial outcomes. Examples of System Blocking Signals are issuance of the "circuit busy" signal to a calling user in the public switched telephone network, and the system's printing of a NET TROUBLE message at a calling operator's terminal in the ARPANET.

test. A process of data extraction that is continuous in time and involves only one factor combination.

transfer denial. A degradation in user information transfer performance defined to occur whenever the performance determined for a transfer sample is worse than the threshold of acceptability for any of the four supported performance parameters as a result of system degradation.

transfer sample. A selected observation of user information transfer performance between a specified source and destination user. A transfer sample includes an integer number of user information blocks, and the inter-block gap that precedes each block. The size of the transfer sample is selected to provide estimates of known precision for four supported performance parameters.

trial. An individual attempt to perform a specified data communication function: access, user information transfer, or disengagement. An equivalent term, frequently used in statistical literature, is "experimental unit."

type I error. The error of rejecting the null hypothesis when in fact it is true. In a typical acceptance test, the Type I error probability expresses the producer's (or service provider's) risk.

type II error. The error of accepting the null hypothesis when it is actually false. The probability of such an error is determined by three variables: the significance level of the experiment, the test sample size, and the actual difference between the hypothetical and "true" values of the tested parameter (in the case where the hypothesis concerns a single parameter).

user blocking signal. A communication signal that informs the system that the issuing user will not participate in a requested data communication session, and the access attempt should therefore be abandoned. The User Blocking Signal is the user's counterpart to a System Blocking Signal. User Blocking Signals are used to identify User Blocking outcomes; these are excluded from system performance measurement. Examples of User Blocking Signals are a calling user's replacing the handset on hook (during connection establishment) in the public switched telephone network, and the called user's issuance of a Close request during a call establishment attempt in the ARPANET.

user information. All information transferred from a source user to the system with the intent that it will ultimately be delivered to a destination user; i.e., all information that is intended to cross both source and destination user/system interfaces. More informally, user information constitutes the "message" the source user wished to convey to the destination.

user information block (block). A contiguous group of user information bits, delimited at the source user interface for transfer to a destination user as a unit. Thus, for instance, a block may be a single ASCII character, a card image, a computer word, or the information field of a frame, depending on the equipment and protocol characteristics of the particular user/system interface. (Note that this definition restricts the content of a "block" to user information bits. Note also that a "block" as defined in this standard is not synonymous with a "transmission block" as defined in American National Standard Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links, ANSI X3.28-1976.)

user/system interface. The functional boundary separating an end user from a data communication system. The standard defines two basic types of user/system interfaces: operator interface and application program interface.

Appendix B

Application Example

This Appendix describes a hypothetical experiment that illustrates how this standard and ANSI X3.102-1983 may be applied. In this example, the two standards are used to measure and evaluate the performance of a data communication service that links user (application) programs in a number of geographically remote host computers.

The example assumes that necessary information about the operation and expected use of the system has been collected. This information is presented at appropriate points in the Appendix. The information does *not* include a detailed description of the internal design of the data communication network, since only an application-program-to-application-program (end-to-end) performance assessment is sought. Therefore, the postulated service could be provided via a packet-switched network, a circuit-switched network, or other network types.

As described in this standard, the measurement process is accomplished in four primary phases:

(1) *Experiment design*. General measurement objectives are developed into a detailed experiment plan. The plan identifies:

(a) The decisions that will be made based on the measurement results

(b) The specific performance parameters to be measured

(c) The users, user/system interfaces, event sequences, and time intervals of interest

(d) The combinations of services, operating modes, traffic levels, or other conditions to be tested

(e) The sampling method to be used in data collection, and the quantities of data to be obtained

(f) The specific conditions to be established during each test

(g) Optionally, a mathematical model to be used in the subsequent data analysis

(2) *Data extraction*. This is the data collection part of the experiment. Signals transferred across selected pairs of digital user/system interfaces are monitored in real time. At each monitored interface, all interface events are detected and their time of occurrence is determined. The observed interface events are mapped into corresponding primary and ancillary reference events as defined in this standard. The reference events and their times of occurrence are recorded. The binary content of the transmitted and received user information is also recorded, where required, to enable the detection of user information error, loss or addition.

(3) *Data reduction*. The primary and ancillary reference events, user information records, and event times observed at the monitored user/system interfaces are merged to associate the data from corresponding interface pairs and data communication sessions. The merged data are processed to produce estimated mean values for the access, user information transfer, and disengagement parameters selected for measurement.

(4) *Data analysis*. The reduced data are examined statistically to determine the precision of the parameter estimates and any associated conclusions. Confidence limits are calculated to quantify the uncertainty of decisions that may be based on the experiment results.

This Appendix is divided into four sections. These describe, successively, the experiment design, data extraction, data reduction, and data analysis phases of the postulated experiment. The forms provided in Section 6 of this standard are used in summarizing the experiment design and data analysis results where appropriate. The description focuses on the experiment design phase since existing procedures and tools are assumed (and referenced) in describing each of the other phases.

B1. Experiment Design

This section describes the design of the hypothetical experiment in terms of the 7-step procedure defined in Section 2 of this standard. Each step is briefly restated, followed by a description of how it might be implemented in the postulated experiment. The entire design is summarized in tabular form at the end of this section.

B1.1 Experiment Objectives. The first step in designing a performance measurement experiment is to define the experiment objectives in a decision context. In this example, it is assumed that a data communication system is being developed to enable information exchange between application programs in host computers at 100 geographically dispersed sites. The data communication system will include:

(1) Communication access software, including the host computer operating systems

(2) Data transmission, switching, and circuit-terminating equipment capable of directly connecting any pair of hosts

Key components of the example are illustrated in Figure B1.

The decision context of the experiment is an acceptance test. The test is planned to determine whether the data communication system meets specified performance requirements. These requirements have been defined in terms of acceptance thresholds for a subset of the parameters described in ANSI X3.102-1983, called the "specified parameters" (see Figure B2.).

The objective of the experiment is to determine whether these acceptance thresholds are met under specified conditions. These conditions include:

(1) A defined data communication session profile, controlled by application programs specifically developed for performance measurement (see B1.3)

(2) Fixed values for key performance factors (see B1.3)

(3) A representative background traffic distribution (see B1.5)

The outcome of the experiment will be used, with the results of other evaluations, in deciding whether to:

(1) Immediately place the computer communication network in full operational service, replacing an existing network

(2) Subject the system to further development and testing prior to such acceptance

Assuming other requirements are met, the users wish to accept the system if the true (population) values for all specified parameters are equal to or better than their acceptance thresholds; and to reject the system otherwise.

The acceptability of the system can be evaluated by a hypothesis test experiment of the type described in the standard. Ideally, the experiment would determine, for each specified parameter, whether the true (population) value is equal to or better than its acceptance threshold, on the one hand, or worse than that threshold, on the other. Because of sampling error, it will be feasible to accurately distinguish these alternatives only when the true value of a specified parameter differs substantially from its threshold. A "region of uncertainty" is therefore defined around each threshold, and the experiment precision objectives are defined in terms of (1) the width of that region and (2) the probability of making incorrect decisions when the true parameter value falls outside the region.

The threshold value and associated region of uncertainty for each specified parameter are used to define the acceptability of all possible population values of that parameter. A population value equal to or better than the high-performance limit of the region of uncertainty is defined as "fully satisfactory"; a population value equal to or worse than the low performance limit of that region is defined as "totally unsatisfactory." A population value within the region of uncertainty is defined as "marginally satisfactory" when it is equal to or better than the threshold, and as "marginally unsatisfactory" otherwise.

The region of uncertainty for each probability parameter extends one-half an order of magnitude in either direction from the threshold value; i.e., if the threshold value is expressed as 10^{-X}, the region of uncertainty is $10^{-(X+0.5)}$ to $10^{-(X-0.5)}$. The region of

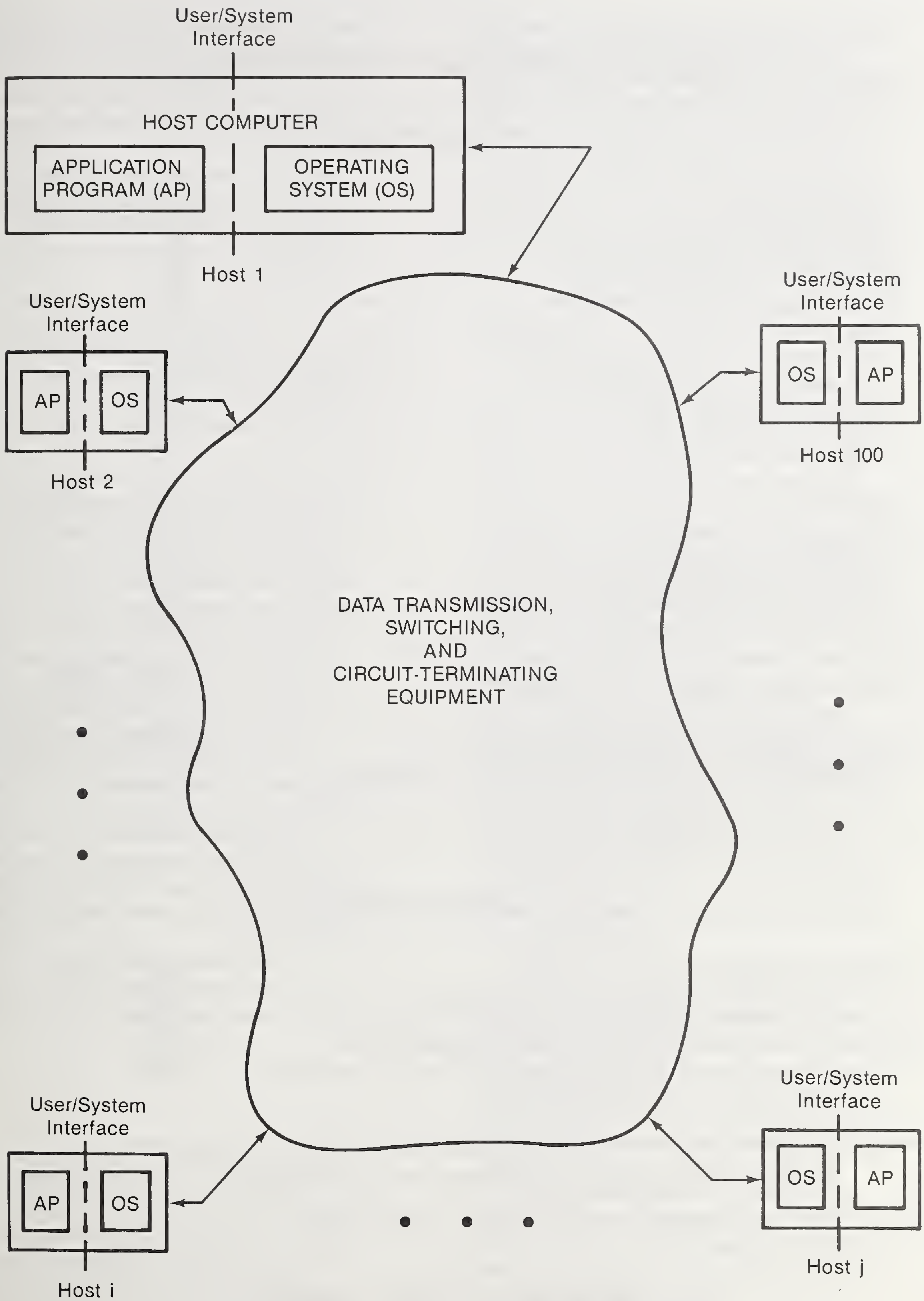


Figure B1
Key Components of the Example

SERVICE PERFORMANCE SPECIFICATION

Part A - Primary Parameters

1. Access Time	12.0	Seconds
2. Incorrect Access Probability	10^{-3}	*
3. Access Denial Probability	10^{-2}	*
4. Access Outage Probability	10^{-2}	*
5. Bit Error Probability	N/A	*
6. Bit Misdelivery Probability	N/A	*
7. Bit Loss Probability	N/A	*
8. Extra Bit Probability	N/A	*
9. Block Transfer Time	2.5	Seconds
10. Block Error Probability	10^{-5}	*
11. Block Misdelivery Probability	N/A	*
12. Block Loss Probability	10^{-5}	*
13. Extra Block Probability	10^{-5}	*
14. User Information Bit Transfer Rate	1000	Bits/ Second
15. Disengagement Time ...(source)	3.0	Seconds
16. Disengagement Denial Probability...(source)	10^{-3}	*
17. Transfer Denial Probability	10^{-3}	*

Part B - Ancillary Parameters

18. User Fraction of Access Time	N/A	*
19. User Fraction of Block Transfer Time	N/A	*
20. User Fraction of Sample Input/Output Time	N/A	*
21. User Fraction of Disengagement Time... (source)	N/A	*

*Note: The probabilities and user performance time fractions are dimensionless numbers between zero and one.

N/A—Not Applicable in this example experiment

Figure B2
Specified Parameter Acceptance Thresholds

uncertainty for each time parameter is bounded by values that are 10% less than and 10% greater than the threshold (e.g., for a delay parameter threshold of 12 seconds, the region of uncertainty is limited by 10.8 and 13.2 seconds).

For each specified parameter, the tested (null) hypothesis can be stated as follows:

The true (population) value of the specified parameter lies in the fully satisfactory region.

The experiment is to be designed to achieve the following precision objectives:

(1) The probability of incorrectly rejecting a system having a fully satisfactory value of a particular parameter is to be 5% or less. The significance level of the hypothesis test is therefore 5%.

(2) The probability of incorrectly accepting a system having a totally unsatisfactory value of a particular parameter is to be 5% or less.

The first precision objective, which refers to a Type I error, can be achieved by accepting the null hypothesis when part or all of the measured 90% confidence interval for the parameter estimate lies in the fully satisfactory region, and rejecting the hypothesis otherwise. The second precision objective, which refers to a Type II error, can be achieved by selecting a sufficiently large sample size, as described in B1.5. The decision criterion defined above is roughly equivalent to accepting the system (with respect to a particular specified parameter) when the estimated parameter value is equal to or better than the threshold, and rejecting the system otherwise.[14]

The probability of accepting the null hypothesis for a particular specified parameter is a function of the population value for that parameter. A typical such function is shown in Figure B3. This curve is called the operating characteristic of the hypothesis test. Figure B3 also indicates the threshold value, the region of uncertainty, and the various categories of acceptability defined earlier.

The users will accept the system if the null hypothesis for each specified parameter is accepted, and will reject the system otherwise. Under the condition that the null hypotheses for the various parameters are independent, the overall probability of accepting the system is the product of the probabilities of accepting the separate null hypotheses.

The overall probability of accepting the system clearly depends on the population values for the individual specified parameters. For example, if the population values of all but one of the specified parameters are far within their fully satisfactory ranges, the probability of accepting the system is approximately equal to the probability of accepting the null hypothesis for the remaining parameter. As indicated in Figure B3, this probability declines from 0.95 to 0.05 as the population value for the parameter varies from the boundary of the fully satisfactory range to the boundary of the totally unsatisfactory range. If the population values for two specified parameters vary over that range, the probability of accepting the system, under the assumption of independence, declines from $(0.95)^2$ or 0.9 to $(0.05)^2$ or 0.0025. The implication for the user is that there is only a small risk (5% or less) of accepting a system for which at least one specified parameter is totally unsatisfactory.

Two other pertinent examples may be noted: (1) the case in which the population value for each of the 12 specified parameters is at the boundary of the fully satisfactory range, and (2) the case in which all population values are at their threshold levels. The probability of accepting the system is approximately $(0.95)^{12}$, or 0.54, in the former case; and only $(0.5)^{12}$, or 2.5×10^{-4}, in the latter case. These examples indicate that the provider should design the system so that the population values of all specified parameters are fully satisfactory, and most are substantially above the boundary of that range.

B1.2 Measured Parameters. The second step in designing a performance measurement

[14] The equivalence is not exact for two reasons: (1) Simply comparing the estimate with the threshold makes no allowance at all for the sampling variation of the estimate; (2) the length of the confidence interval achieved in the test usually differs from the expected length used in the test design. For example, a parameter estimate may lie on the low performance side of the threshold when the confidence interval extends into the fully satisfactory region. In such a situation, the system will be accepted (with respect to the parameter in question) on the basis of the confidence interval, but would be rejected on the basis of the parameter estimate.

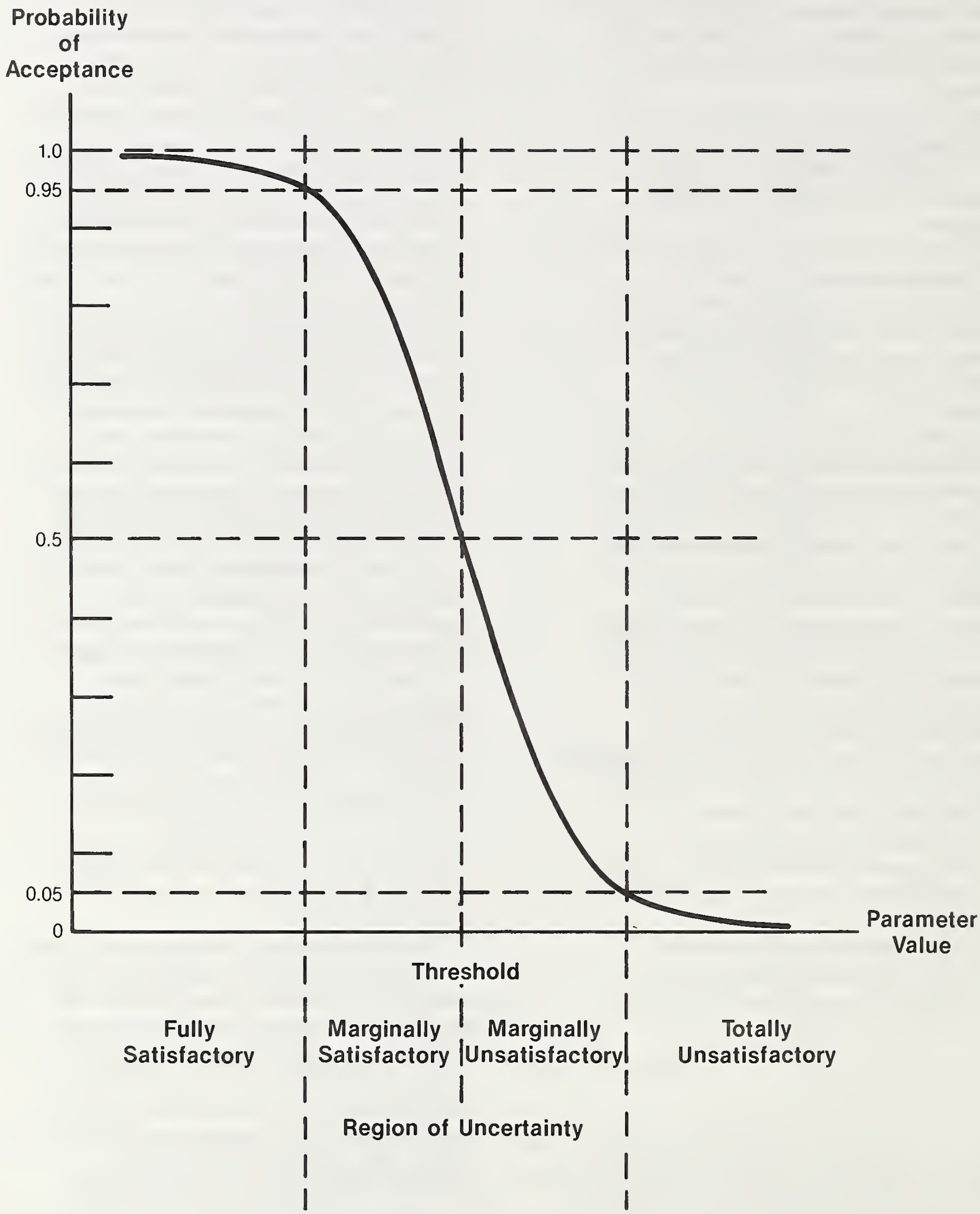


Figure B3
Acceptance Criteria

experiment is to select the particular parameters to be measured. That selection is straightforward in the postulated experiment. All parameters for which acceptance thresholds have been defined (e.g., all "specified" parameters) must be measured, since their measured values will directly influence the decision to accept or reject the system. Parameters for which acceptance thresholds have not been defined need not be measured, since their measured values will have no such influence. As shown in Figure B2, acceptance thresholds have been defined for all parameters except the bit-oriented primary parameters (parameters 5-8) and the ancillary parameters (parameters 18-21). The bit-oriented primary parameters are not of direct interest in this experiment because the user application programs process only complete blocks. The ancillary parameter values are determined by their corresponding primary parameter values, since the total user delays in performing each function are fixed by the application program designs (see Section B2).

B1.3 Population Definition. The third step in designing a performance measurement experiment is to define the population of performance trials (e.g., call attempts) on which the measurements will focus (and to which the measured values will refer). This is a key step in the experiment design process. Seven items of information that must be specified to fully define an experiment population are identified in 2.3 of this standard. Their specification in the postulated experiment is summarized in the following paragraphs.

(1) Characteristics of the user pairs to which the data communication service is provided.

To facilitate control and improve repeatability, existing data extraction software, described in [B1],[15] is used in the present experiment. This existing software consists of two programs. The first, a program called XMIT, performs necessary end user activities and also records the nature and time of occurrence of interface signals in the computer selected as source (transmitting) host during a test. The second, a program called RECV, performs the corresponding functions in the selected destination host. Thus, XMIT and RECV are the end users in this experiment. Their functions and performance in the various host computers are nominally identical. These test programs are briefly described in Section B2.

(2) Overall set of user pairs to be represented (as distinguished from the subset actually measured).

The example involves 100 geographically dispersed host computers, each potentially containing an XMIT and a RECV program. The associated data communication system is designed to be capable of directly interconnecting any pair of hosts, so that the overall set of user pairs to be represented consists of all (100)(99) or 9900 possible XMIT/RECV pairs. Obviously, a much smaller number of user pairs will actually be tested (see B1.5).

(3) Observation period (or periods) within which performance is to be characterized (and within which tests may be conducted).

The specified performance requirements must be satisfied during the busy period of the day. The host sites are geographically dispersed across the four continental United States time zones. With respect to locally generated traffic, each host experiences a busy period from 3 p.m. to 5 p.m. local time, Monday through Friday. Each host also experiences a busy period resulting from traffic generated during the local busy periods in the other time zones. The result is an extended busy period for each host. The extended busy period begins at the start of the local busy period in the Eastern time zone, and ends at the end of the local busy period in the Pacific time zone. This extended (5 hour) busy period is expressed in local time in each of the four time zones as follows:

[15] Numbers in brackets refer to corresponding numbers in Section B5, "References."

Host Computer Location (U.S. Time Zone)	Busy Period at Host Site
Eastern	3 p.m. - 8 p.m. EST
Central	2 p.m. - 7 p.m. CST
Mountain	1 p.m. - 6 p.m. MST
Pacific	12 p.m. - 5 p.m. PST

All tests will be conducted during the extended busy period.

(4) Characteristics of the user/system interfaces to be monitored in collecting the performance data, including the placement of measurement points.

Figure B4 illustrates the user/system interfaces, and interface events, to be monitored in the postulated experiment. In each host computer, the user/system interface will be defined to correspond to the software functional interface between the test application program (XMIT or RECV) and the computer's operating system. Signals communicated across these interfaces will be of two general types: **system calls**, which are issued by an application program to request the performance of a particular operating system function; and **system responses**, which are issued by the operating system to indicate completion of a previously requested function.

In this example, the four operating system calls that may be issued by application programs executing in the selected host computers are OPEN, WRITE, READ, and CLOSE. With respect to remote communication, OPEN and CLOSE are used to establish and release connections between application programs. WRITE causes a specified block of user information to be passed from a source application program to its local operating system for transmission over an established connection. READ causes received user information to be passed from the receiving operating system to the local (requesting) application program. In the normal case, the system's "complete" responses indicate that the requested function has been successfully accomplished. System failures are indicated by special exception codes. Each system call and response will be recorded and time-stamped as discussed in Step (6).

(5) Session profiles (or equivalent descriptions) defining the interface event sequence(s) that occur at the monitored user/system interfaces during a typical (successful) data communication session of the type to be observed; and any service refusal (e.g., blocking) or service interruption (e.g., preemption) sequences explicitly allowed by the system design.

Figure B5 illustrates the sequence of interface events that occurs at the monitored user/system interfaces during a normal (successful) data communication session in the postulated experiment. As shown in the legend, the solid arrows represent interface signals and the dashed arrows indicate cause/effect relationships between signals. The functions of these interface signals have been explained earlier with the exception of the OPEN (ANY HOST) request; that signal notifies the system of RECV's willingness to begin a data communication session with any user.

Abnormal events considered consist of negative responses to the OPEN, READ, and WRITE requests and non-responses to all signals. In the example, it is assumed that the XMIT and RECV programs report negative responses to a test monitor, who then has the choice of terminating or restarting the test. The test monitor function may be manual or automated, and may be performed at the local test site or a remote site. The test monitor will restart the test in the case of nonresponses (performance timeouts).

(6) Reference events corresponding to each interface event defined in the session profile(s). Data communication session type (connection-oriented or connectionless).

Relationships between the user/system interface signals observed during the hypothetical test session and the reference events defined in this standard are indicated in Figure B5 by paired event numbers below each interface signal. The first number in each pair identifies the primary reference event to which the interface signal corresponds; the second number identifies the ancillary event. The primary reference events are listed in Table 3 of this standard; the ancillary events are listed in Table 5. In the former case, the digit 0 is used to indicate no significance; i.e., the signal has no corresponding primary reference event.

The OPEN (ANY HOST) request issued by the RECV program has two performance effects. First, it commits the RECV program to participate in a future data communication session. Second, it temporarily relieves both RECV and the system of responsibility for generating a subsequent event at the local (RECV) interface, since the next event in sequence (XMIT issuance of an OPEN request) necessarily occurs at the remote (XMIT) interface. The OPEN (ANY HOST) request has no remote responsibility effect. These performance effects are represented by the primary/ancillary reference event pair (2/3): Nonoriginating User Commitment/local responsibility undefined, no remote effect. Note that the OPEN (ANY HOST) request does not initiate a data communication session or begin the timing of an access trial. It does, however, identify the data communication session as a connection-oriented session.

The OPEN request issued by the XMIT program has three performance effects. First, it initiates the data communication session and starts the timing of an access trial. Second, it makes the system responsible for generating the next event at the local (XMIT) interface -- the OPEN COMPLETE response. Finally, it makes the system responsible for generating a similar OPEN COMPLETE response at the remote (RECV) interface. These performance effects are represented by the primary/ancillary reference event pair (1/4): Access Request/local system responsibility, remote system responsibility.

The OPEN COMPLETE responses have no primary performance effects, leave the user responsible for creating the next event at the local interface, and have no remote responsibility effects. Their occurrences are therefore recorded, in each case, by the reference event pair (0/2).

The WRITE requests issued by XMIT have the same ancillary performance effects as the OPEN request. Each WRITE request is an instance of both primary events 5 and 6 (Start of Block Input to System and Start of Block Transfer), which are coincident in this application. The first WRITE request in a data communication session stops the timing of the access trial. Each WRITE request starts the timing of a block transfer trial.

The READ requests issued by RECV have no primary performance effects, but they do temporarily relieve both RECV and the system of responsibility for generating a subsequent event (it is assumed that each READ request is issued prior to the counterpart WRITE request). They are represented by the primary/ancillary event pair (0/3). The WRITE COMPLETE signals issued to XMIT by the system have the same performance effect as the OPEN COMPLETE response (0/2). READ COMPLETE signals communicating a nonzero byte count are End of Block Transfer events, leave the user responsible for creating the next event at the local interface, and have no remote responsibility effects. They are represented by the reference event pair (7/2). Each such event stops the timing of a block transfer trial.

The first CLOSE request issued by a user is the Disengagement Request (and starts the timing of disengagement) for *both* participating users (8/4). The last READ COMPLETE signal (communicating a zero byte count) has the same performance effect as the OPEN COMPLETE response (0/2). The subsequent CLOSE request has no primary performance effect, but transfers responsibility to the system. Each CLOSE COMPLETE signal confirms disengagement and stops the timing of the disengagement trial for the receiving user. The last CLOSE COMPLETE signal terminates the data communication session.

(7) Timeouts and thresholds that distinguish successful trials from performance failures (as defined in ANSI X3.102-1983).

Timeouts are specified for the access, block transfer, and source disengagement functions and for transfer sample input/output. In all four cases, the timeout duration is 3 times the specified performance time. The specified performance times for the access, block transfer, and source disengagement functions are given in Figure B2, and the corresponding timeout durations are given in Table B1. The transfer sample input/output timeout can be determined as follows:

(1) Determine the Transfer Denial threshold for User Information Bit Transfer Rate:
1000 bps/3 = 333 bps

(2) Determine a minimum transfer sample size

In ANSI X3.102-1983 it is required that a transfer sample contain sufficient bits to enable each of 4 supported performance parameters to be estimated (at their threshold values) with a relative precision of 50% and a confidence level of 95%. In this application, the specified value for Block Error Probability dictates the minimum transfer sample size. If only 1 incorrect bit is contained in every Incorrect Block, and a user information block length of 512 bits is specified, the Bit Error Probability value corresponding to a Block Error Probability of 10^{-4} is $10^{-4}/512$ or 2×10^{-7}. In accordance with ANSI X3.102-1983, the corresponding Transfer Denial threshold is $(2 \times 10^{-7})^{1/4}$ or 2.1×10^{-2}. Eighteen bit errors must be observed to estimate a failure probability with 50% relative precision and a 95% confidence level [B2]. The minimum transfer sample size can therefore be calculated as follows:

$$\frac{\text{18 bit errors}}{2.1 \times 10^{-2} \text{ bit errors/transferred bit}} = \text{857 bits}$$

It is decided to make the transfer sample consist of 3 blocks, and a transfer sample size of 1536 bits is therefore selected.

(3) Calculate the transfer sample input/output timeout as follows:

$$\begin{aligned}\text{Transfer Sample Input/Output Timeout} &= \frac{\text{Selected Transfer Sample Size}}{\text{User Information Bit Transfer Rate Threshold}} \\ &= \frac{\text{1536 bits}}{\text{333 bits/second}} \\ &= \text{4.6 seconds.}\end{aligned}$$

The user performance time fractions shown in Table B1 are calculated by dividing the (fixed) user performance times for each function by the corresponding specified time parameter values (Figure B2). These are used to identify the entity responsible for timeout performance failures.

B1.4 Performance Factors. The fourth step in designing a performance measurement experiment is to specify the factors presumed to influence performance, the relevant levels or values of each factor, and the factor combinations to be tested. The postulated experiment is a simple hypothesis test involving only one level for each factor and only one factor combination. The test conditions can therefore be specified by simply stating the selected level of each relevant performance factor. Key system design features such as topology, protocols, and transmission and switching technologies would normally be specified, but are omitted in this hypothetical example. Performance factors that influence the example experiment design are listed below.

Performance Factor	Level
Access Line Speed	1200 bps
User Information Block Size	512 bits
Time of Day/Week	Busy Period/M-F

User performance times are fixed by the XMIT and RECV program designs, as discussed earlier.

B1.5 Population Sample. The fifth step in designing a performance measurement experiment is to select from the defined population a representative sample of performance trials (e.g., access attempts) to be measured. This requires that decisions of two types be made:

(1) *Primary decisions*, that are critical to achieving the experiment precision objectives. These decisions are the number of user pairs selected for testing, the total number of performance trials to be observed, and the number of trials per tested user pair.

(2) *Secondary decisions*, that influence the experiment precision less strongly and therefore may be made on the basis of convenience. These decisions are the number of access/disengagement and block transfer tests conducted on each selected user pair, the number of data communication sessions per test, and the

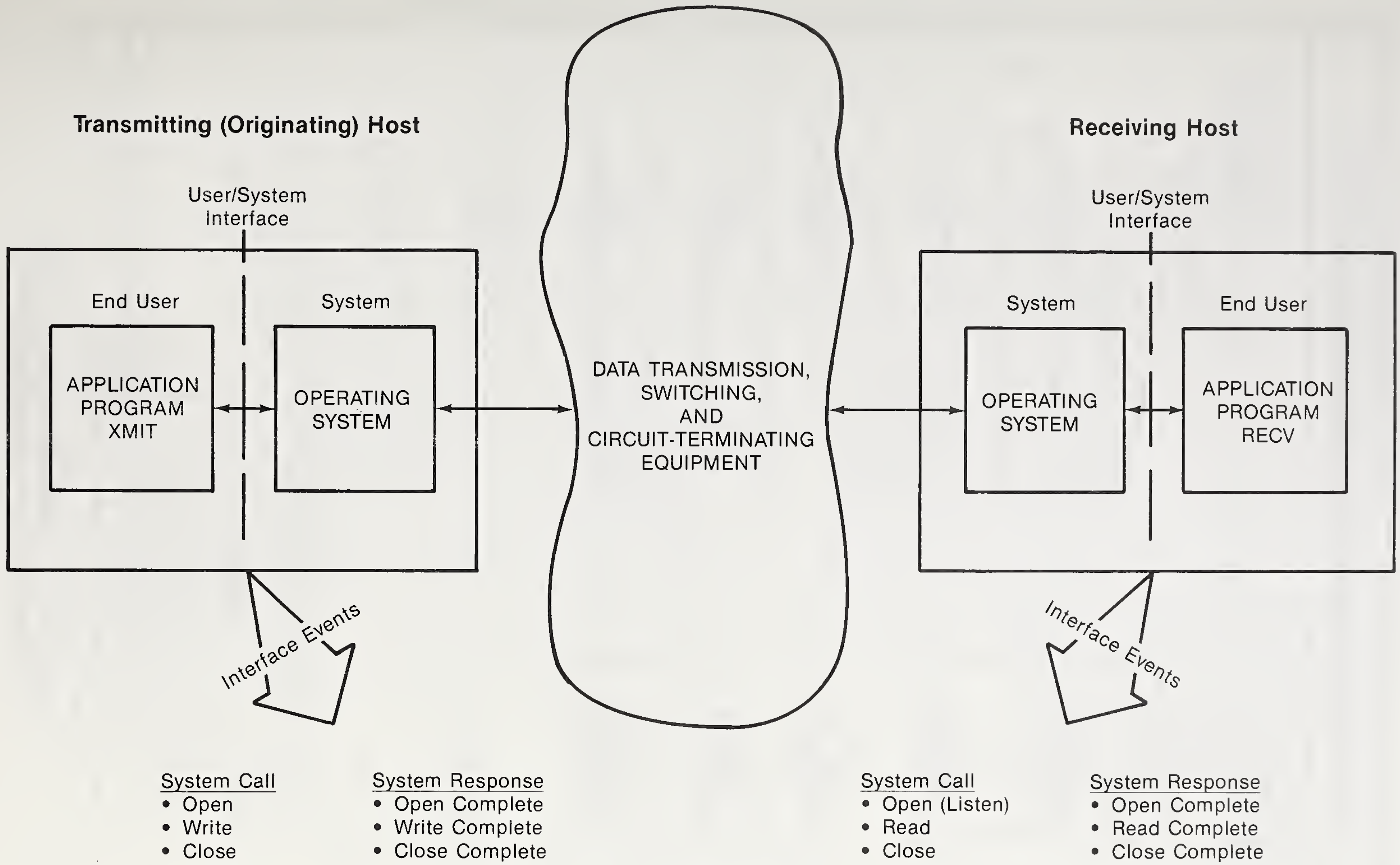
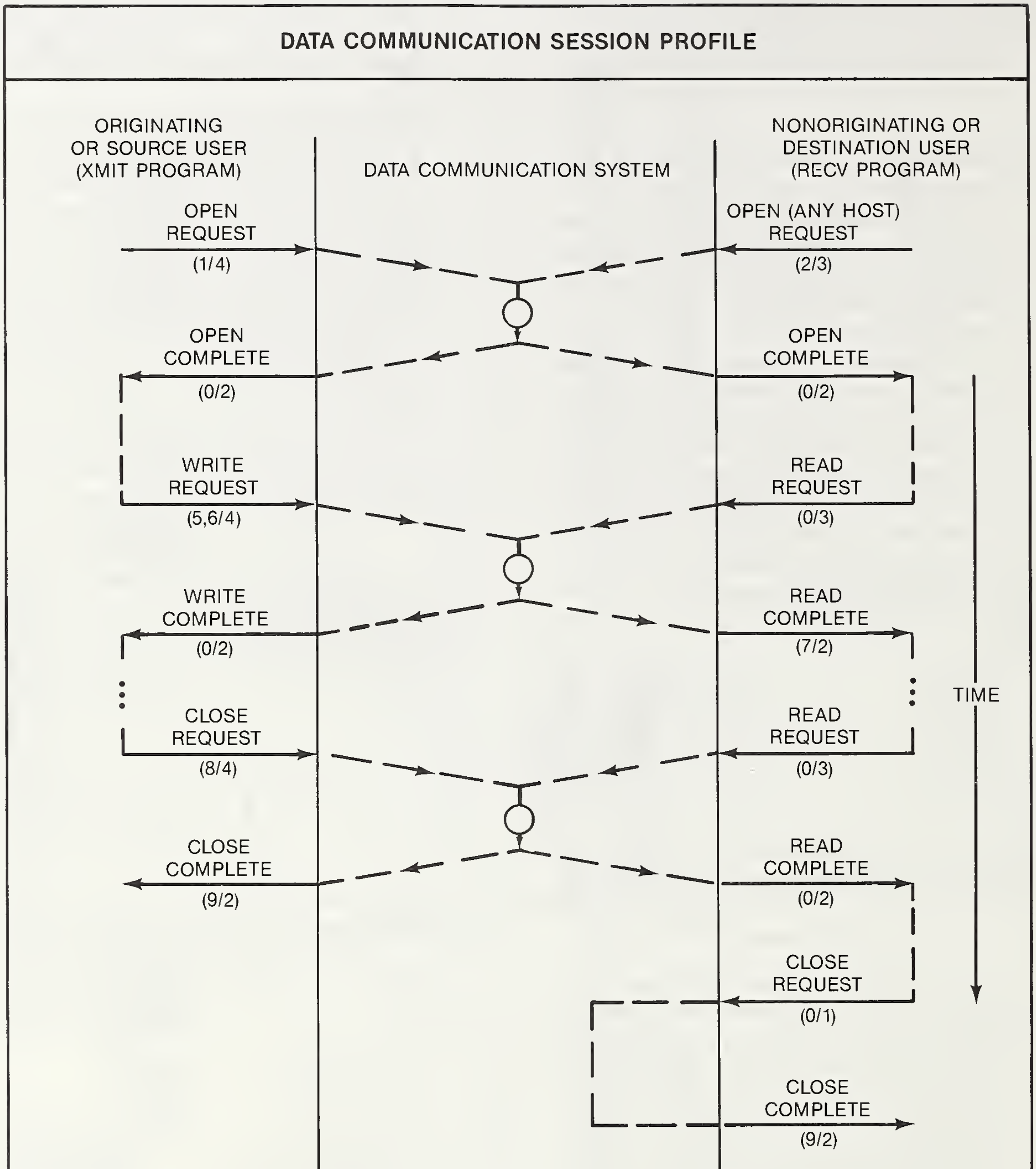


Figure B4
User/System Interface Characteristics

Successful Session



Notes: 1. Primary event 5 (start of block input to system) and primary event 6 (start of block transfer) are coincident in this application.
2. The digit 0 identifies interface events that have no corresponding primary reference event.

LEGEND

(NAME) →	Interface Signal (User Input to System)
(NAME) ←	Interface Signal (System Output to User)
0/1 • • • (9/6)	Reference Event Number (Primary/Ancillary)
— — —	Cause/ Effect Linkage Between Signals

Figure B5
Experiment Data Communication Session Profile

PERFORMANCE TIMEOUT AND THRESHOLD SPECIFICATIONS	
1. Performance Timeouts	
a. Access	36.0 sec.
b. User Information Block Transfer	7.5 sec.
c. Transfer Sample Input/Output	4.6 sec.
d. Source User Disengagement	9.0 sec.
e. Destination User Disengagement	N/A
2. User Performance Time Fractions	
a. Access	0.15
b. User Information Block Transfer	0.13
c. Transfer Sample Input/Output	0.13
d. Source User Disengagement	0
e. Destination User Disengagement	N/A
3. Transfer Denial Criteria	
a. Transfer Sample Size	3 Blocks
b. Bit Error Probability Threshold	2.1×10^{-2}
c. Bit Loss Probability Threshold	N/A
d. Extra Bit Probability Threshold	N/A
e. User Information Bit Transfer Rate Threshold	333 bps

N/A—Not Applicable in this experiment.

Table B1
Timeout and Threshold Specifications for the Experiment

number of transmitted user information blocks per session.

As noted earlier, the total number of user pairs interconnected by the network (9900) is far too great to enable the testing of all user pairs. In this example, a much smaller number of user pairs is selected for testing on the basis of the network hierarchy and corresponding traffic variation.

The postulated computer communication network is organized into seven regions. Each region has one computer that is designated as the regional computer. The other computers are designated as local computers within their regions, and as "remote" computers in describing connections with computers in other regions. The number of computers located in each region is:

Region	Computers
1	20
2	11
3	19
4	15
5	10
6	8
7	17

Five types of computer-to-computer connections are distinguished, as illustrated in Figure B6. The numbers of user pairs and the proportions of traffic corresponding to each connection type are summarized in the following table:

Connection Type	Number of User Pairs	Traffic (%)
Local to regional	186	35
Local to local	1274	25
Regional to regional	42	20
Remote to regional	1116	15
Remote to remote	7282	5
	9900	100

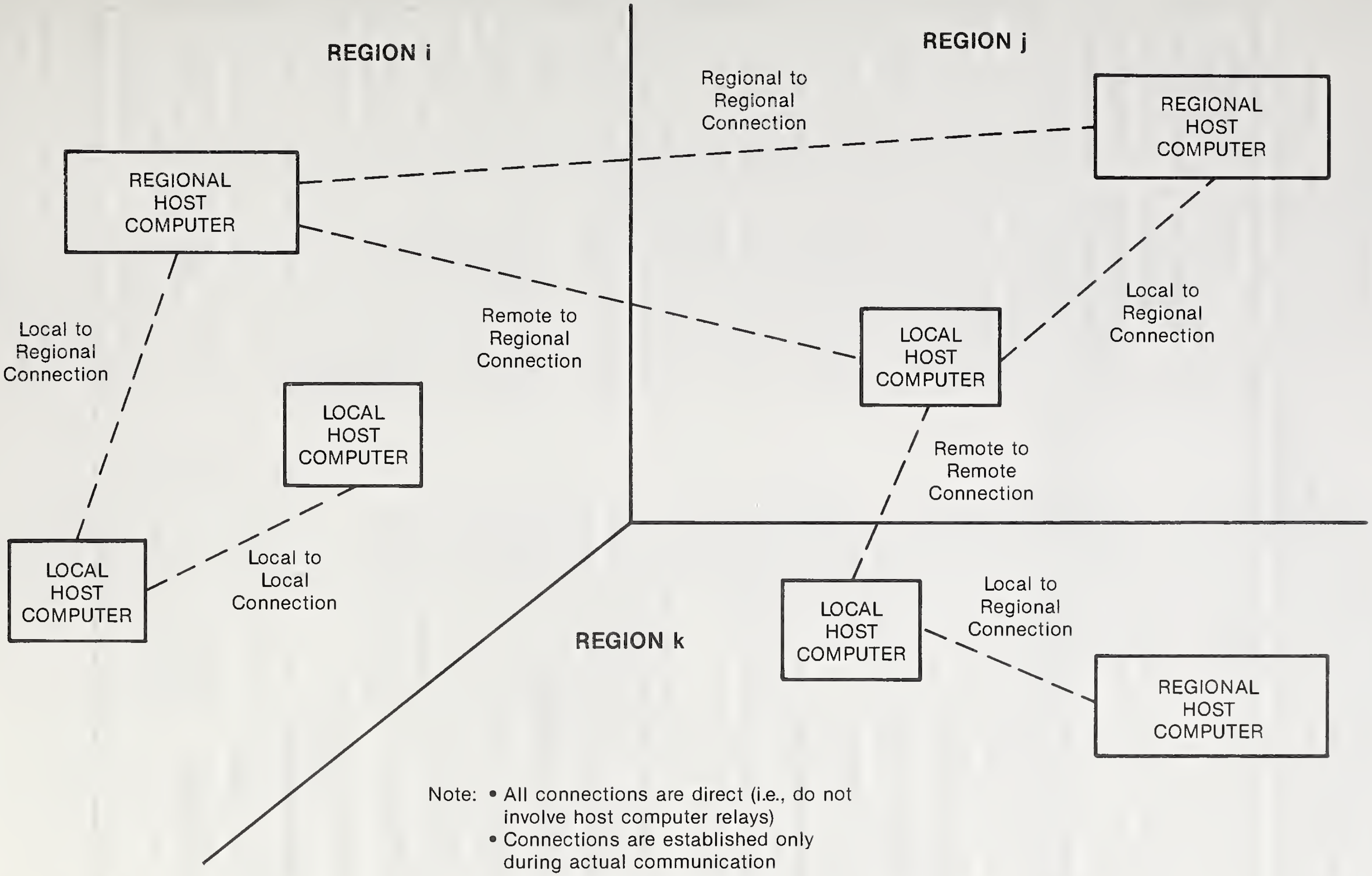
The numbers of user pairs and proportions of traffic differ substantially among connection types, and purely random sampling is therefore not appropriate. Since 60% of the traffic falls into the Local-Local and Local-Regional connection types, each of these connection types will be sampled randomly within each of the seven regions. The Regional-Regional connection type involves sufficient traffic so that the sample will include each region once in the set of user pairs measured. The tested Regional-Regional connections will be selected randomly within the above constraint. The Remote-Regional and Remote-Remote connection types carry relatively little traffic; each will be tested by selecting user pairs randomly from the overall set without regard to their balance among regions.

The optimum number of user pairs to be tested is difficult to determine quantitatively. Selecting too few user pairs can reduce the effective sample size by increasing the correlation between trials. Selecting a very large number of user pairs increases the cost and complexity of the measurements. Generally, preliminary experiments are not helpful because the necessary preliminary sample size is a large fraction of the overall experiment sample size.

A good rule of thumb is to have between 10 to 30 user pairs in a sample. The exact choice depends upon the available test time and budget. In the present experiment, it is decided that the Local-Regional and Local-Local groups will each be represented by 7 user pairs, randomly selected such that each region will have one user pair of each type in the sample. The Regional-Regional, Remote-Regional, and Remote-Remote groups will be represented by 4, 3, and 3 randomly selected user pairs, respectively. The number of user pairs to be measured in sampling the 9900 possible user pairs is therefore 24.

This experiment requires measurements to be made on three time parameters, one rate parameter, and eight failure probabilities. The measurements involve all three primary data communication functions: access, user information transfer, and disengagement. In accordance with the design of the existing XMIT and RECV programs, the experiment will be organized into two types of tests. One type of test will measure all specified access and disengagement parameters. The other type of test will measure all specified user information transfer parameters. This organization of the experiment will improve the efficiency of the measurement process.

A single sample size will be selected for each type of test. Each sample size will be chosen so as to achieve the most stringent precision requirement -- in this example, the precision requirement for the lowest failure probability among the set of parameters to be



Note:
- All connections are direct (i.e., do not involve host computer relays)
- Connections are established only during actual communication

Figure B6
Computer Network Hierarchy and Connections Types

Table B2
User Pair Sample Sizes

HOST COMPUTER CONNECTION	NUMBER OF USER PAIRS TESTED	NUMBER OF ACCESS/ DISENGAGEMENT PERFORMANCE TRIALS	NUMBER OF USER INFROMATION BLOCK TRANSFER PERFORMANCE TRIALS
1. Local-Local Region 1	1	60	1000
2. Local-Local Region 2	1	60	1000
3. Local-Local Region 3	1	60	1000
4. Local-Local Region 4	1	60	1000
5. Local-Local Region 5	1	60	1000
6. Local-Local Region 6	1	60	1000
7. Local-Local Region 7	1	60	1000
8. Local-Regional Region 1	1	60	1000
9. Local-Regional Region 2	1	60	1000
10. Local-Regional Region 3	1	60	1000
11. Local-Regional Region 4	1	60	1000
12. Local-Regional Region 5	1	60	1000
13. Local-Regional Region 6	1	60	1000
14. Local-Regional Region 7	1	60	1000
15. Regional-Regional	4	240	4000
16. Remote-Regional	3	180	3000
17. Remote-Remote	3	180	3000
	24	1440	24,000

measured. The precision requirements for all other parameters in each set will then be exceeded.

The lowest failure probability specified for the access and disengagement parameters is 10^{-3}. Based on the specified confidence limits of 3.16×10^{-4} to 3.16×10^{-3}, and the specified Type I and Type II error probabilities of 5%, a minimum sample size of 1122 (or about 1200) trials is required (see [B3, Table 17, p. 225]). This sample size is expected to be more than adequate to achieve the desired precision in measuring the two access failure probability parameters that are specified at 10^{-2}, as well as the access and disengagement times. The latter assumption could be verified in a preliminary experiment.

The three block transfer failure probabilities specified in Figure B2 are each 10^{-4}. Based on the specified confidence limits of 3.16×10^{-5} to 3.16×10^{-4} and the specified Type I and Type II error probabilities of 5%, a minimum sample size of 11 230 (or about 12 000) trials is required (see [B3, Table 17, p. 225]). Hence, at least 12 000 blocks must be transmitted in the experiment to estimate the block error, block loss, and extra block probabilities to within one-half an order of magnitude. This sample size is expected to be more than adequate for measurement of Block Transfer Time and User Information Bit Transfer Rate.

Twelve hundred access/disengagement trials distributed evenly among 24 user pairs represent 50 trials per pair. Similarly, 12 000 block transfer trials distributed evenly among 24 user pairs represent 500 trials per pair. Although these minimum sample sizes should be sufficient to achieve the experiment objectives, it is relatively easy to obtain and process more trials once the XMIT and RECV programs have been installed in a particular pair of host computers. To allow for defective test data (often 10 to 30% of the planned data in practical experiments) and to provide higher confidence that the desired measurement precision will be achieved, it is decided to increase the number of access/disengagement trials per user pair to 60, and the number of block transfer trials per user pair to 1000. The resulting total numbers of access/disengagement and block transfer trials are 1440 and 24 000, respectively. These primary sampling decisions are summarized in Table B2.

As noted earlier, three secondary sampling decisions must also be made. These are (1) the number of access/disengagement and block transfer tests conducted on each selected user pair, (2) the number of data communication sessions per test, and (3) the number of transmitted user information blocks per session.

To simplify the data consolidation and reduction processes, it is decided that the 60 access and disengagement trials conducted on each selected user pair will be grouped in six tests, with 10 data communication sessions (i.e., 10 access trials and 10 disengagement trials) nominally contained in each test. The tests will be conducted in two successive weeks. Each test will be initiated at the beginning of a randomly selected busy period; and XMIT will be programmed to initiate successive sessions at roughly 1/2 hour intervals throughout the remainder of the busy period. Thus, each access/disengagement test will sample almost uniformly across a selected busy period. A single 512-bit user information block will be transmitted during each data communication session to verify successful access.

A similar approach is followed in grouping the block transfer trials. It is decided to conduct 10 tests per user pair, with one hundred 512-bit user information blocks transmitted (in a single data communication session) during each test. Each test will be initiated at a randomly selected time during a randomly selected busy period to avoid possible bias. Each test will last approximately 1 minute assuming the 1000 bps throughput objective is met. Table B3 summarizes the sample size decisions.

Table B3
Sample Size Summary

Parameters	Number of User Pairs	Tests per User Pair	Trials per Test	Total Trials
Access	24	6	10	1 440
Block Transfer	24	10	100	24 000
Disengagement	24	6	10	1 400

In each test, a single user will serve both as the session originator and the source of transferred user information.

B1.6 Test Conditions. The sixth step in designing a performance measurement experiment is to specify the particular factor combination to be used in each test. The postulated experiment is a simple hypothesis test involving only one factor combination. Therefore, this factor combination (the set of factor levels specified in B1.4) will be used in all tests.

B1.7 Mathematical Model. The seventh (optional) step in designing a performance measurement experiment is to summarize the experiment design and the planned data analysis in an explicit mathematical model. Formulas for combining individual observed values (e.g., individual access times) to estimate the corresponding parameter values (e.g., access time sample means) are provided in Section 3 of this standard. The only random variable in the model is the experimental error, which is taken into account by calculating confidence limits for each parameter for a prescribed confidence level. As outlined in B1.1, the purpose of the experiment is to test hypotheses about the parameters relative to their threshold values, but the design was accomplished using the equivalent confidence interval approach. Formulas for calculating the experimental error in the parameter estimates (i.e., confidence limits) are provided for each type of parameter in [B4]. Confidence limits used for experimental error estimation and hypothesis testing were computed based upon the use of:

(1) The binomial distribution and its Poisson approximation for probability parameters such as Block Error Probability

(2) The Gaussian (normal) time distribution for parameters such as Access Time

Each of the seven steps in the design of the hypothetical experiment has now been described. The entire design is concisely summarized in Table B4.

B2. Data Extraction

This section describes the data extraction phase of the postulated experiment. Three major functions are accomplished during this phase:

(1) The XMIT and RECV programs are installed in the selected host computers.

(2) A series of 16 tests are conducted, using these programs, on each selected user pair (6 access/disengagement tests and 10 block transfer tests).

(3) The data extracted during all tests are consolidated in one computer and prepared for processing by the data reduction programs.

Figure B7 shows the normal flow of the XMIT and RECV programs and the interactions between them. During each test, the two programs progress in synchronism through three consecutive phases of operation: A pre-exchange phase, associated with connection establishment or access; an exchange phase, associated with actual user information transfer; and a post-exchange phase, associated with connection release or disengagement.

The XMIT and RECV programs are designed to perform either of two types of tests: an access/disengagement test and a user information transfer test. The numbers of data communication sessions and performance trials included in each test (and the delays, if any, between them) are selected by the test operator. As specified in B1.5, each access/disengagement test comprises 10 data communication sessions, to be initiated at half-hour intervals over a 5-hour period. A single 512-bit block is transmitted during each session. Each user information transfer test comprises a single data communication session, in which one hundred 512-bit blocks are transmitted with no intervening delays.

Correlation of event times requires accurate synchronization of clocks in the geographically remote computers hosting each user pair. This is accomplished, in the postulated experiment, through the use of a National Bureau of Standards (NBS) time dissemination service provided via the National Oceanic and Atmospheric Administration's GOES satellite. The NBS time dissemination service is described in

[B6]. Its use in actual performance measurements is described in [B1].[16]

Data extracted in the various tests are consolidated in one computer (a Regional computer selected for data reduction) using available network file transfer programs. To ensure the integrity of the file transfer process, the files are protected by retransmission error control. The consolidated data are converted to a specified ASCII character format (described in [B1]) to prepare them for data reduction.

A detailed description and listing of the XMIT and RECV computer programs selected for use in this postulated experiment are provided in [B1]. The programs are written in "C" language and are executable on many 16-bit microcomputers. Off-line "C" language programs that consolidate the extracted data, perform time correction, generate the necessary ASCII character files, and produce specialized measurement results such as delay histograms are described in the same reference.

B3. Data Reduction

This subsection describes the data reduction phase of the postulated experiment. In this phase, a set of data reduction computer programs processes the performance data collected during the data extraction phase to produce estimated mean values for the specified performance parameters. Although the experiment is hypothetical, the description refers to existing data reduction computer programs that are available to users of this measurement standard. These programs were originally written in ANSI FORTRAN 66[17] and are adaptable for use on most general purpose computers having a word length of 16 or more bits. The programs are described in more detail in [B1].

Figure B8 outlines the data reduction scheme used in the postulated experiment. The reduction is accomplished by two sets of reduction runs. One set processes performance data from access/disengagement tests, and the other processes data from block transfer tests. Each reduction run processes the performance data from a single test, and is executed off-line after all data for the test have been recorded and consolidated in one computer.

For each reduction run, the specifications input file contains identifiers that characterize the particular run and information used to control various reduction procedures. The latter includes the test type (which determines the set of performance parameters to be evaluated), and the performance timeout and threshold specifications used by reduction routines in classifying the outcomes of individual performance trials. The performance data batch for the run consists of a set of files containing the source and destination reference event histories recorded by the data extraction programs during a particular test, as described in [B1].

Data processing is carried out by a sequence of three FORTRAN programs, each of which implements a distinct phase of the overall reduction process. PROLOG is the first program executed in a reduction run. It carries out preliminary validation and merging of the input data. PROLOG first reads the specifications input and performance data files and subjects their contents to a series of validity checks. If no errors are detected, PROLOG then combines the source and destination reference event data to create a unified event history.

ANALYZ, the second program executed in a reduction run, is responsible for performance data assessment and parameter calculation. It begins by reading the specifications file generated by PROLOG and by initializing relevant performance statistics (outcome counts and cumulative performance times) to zero. ANALYZ then examines the reference event records to identify individual performance trials and classify their outcomes. As outcomes are determined, ANALYZ updates the affected performance statistics and records each outcome in the appropriate performance outcome file for input to the data analysis

[16]This service makes it possible to obtain a time signal accurate to within 1 ms anywhere in North America. Several vendors supply a clock receiver unit to obtain time from the satellite. The receiving antenna is 1 foot square and operates satisfactorily inside many buildings. The clock unit is provided with a standard RS-232-C interface. The cost of the unit is about $2,500.

[17]The programs are being updated to ANSI FORTRAN 77. For current information, please contact the X3 Secretariat.

Table B4
Experiment Design Summary

(1) **Experiment Objectives:** Acceptance test (hypothesis test experiment). Tested hypothesis (for each of 12 specified parameters): true (population) value is within a defined "fully satisfactory" region.

(2) **Measured Parameters:** Access Time, Incorrect Access Probability, Access Denial Probability, Access Outage Probability, Block Transfer Time, Block Error Probability, Block Loss Probability, Extra Block Probability, User Information Bit Transfer Rate, Disengagement Time (source), Disengagement Denial Probability (source), Transfer Denial Probability.

(3) **Population Definition:**

(a) *User Pair Types*. XMIT and RECV test computer programs designed to emulate real application programs in the network host computers.
(b) *User Pairs Represented*. 9900 possible combinations of XMIT and RECV programs in the 100 host computers.
(c) *Observation Period(s)*. Five-hour busy period encompassing 3-5 p.m. local time in each of the four U.S. time zones.
(d) *User/System Interface Characteristics*. Application program/operating system interface providing OPEN, WRITE, READ, and CLOSE system calls and associated complete responses.
(e) *Session Profile*. See Figure B5.
(f) *Reference Events (and Session Type)*. See Figure B5. Session is connection oriented.
(g) *Timeouts/Thresholds*. See Table B1.

(4) **Performance Factors:** Experiment is a simple hypothesis test involving only one factor combination:

Performance Factor	Level
Access Line Speed	1200 bps
User Information Block Size	512 bits
Time of Day/Week	Busy Period/M-F

(5) **Population Sample:**

(a) *Method of Selection*. Traffic-weighted random sampling of five connection types. Equal representation among seven regions.
(b) *Sample Size*. 1 440 access/disengagement trials, 24 000 block transfer trials. See Tables B2 and B3.
(c) *Confidence (or significance) Level*. Probability of incorrectly rejecting a system having a fully satisfactory value of a parameter (Type I error probability) 5% or less. Probability of incorrectly accepting a system having a totally unsatisfactory value of a parameter (Type II error probability) 5% or less.

(6) **Test Conditions:** Factor combination in (4) above used in all tests.

(7) **Mathematical Model:** Random variable considered is experimental error -- represented by confidence limits/levels. Probability parameters: range of uncertainty about each threshold ± 1/2 order of magnitude; binomial distribution (Poisson approximation) used in calculating confidence limits. Time Parameters: Range of uncertainty about each threshold ± 10%; Gaussian distribution used in calculating confidence limits.

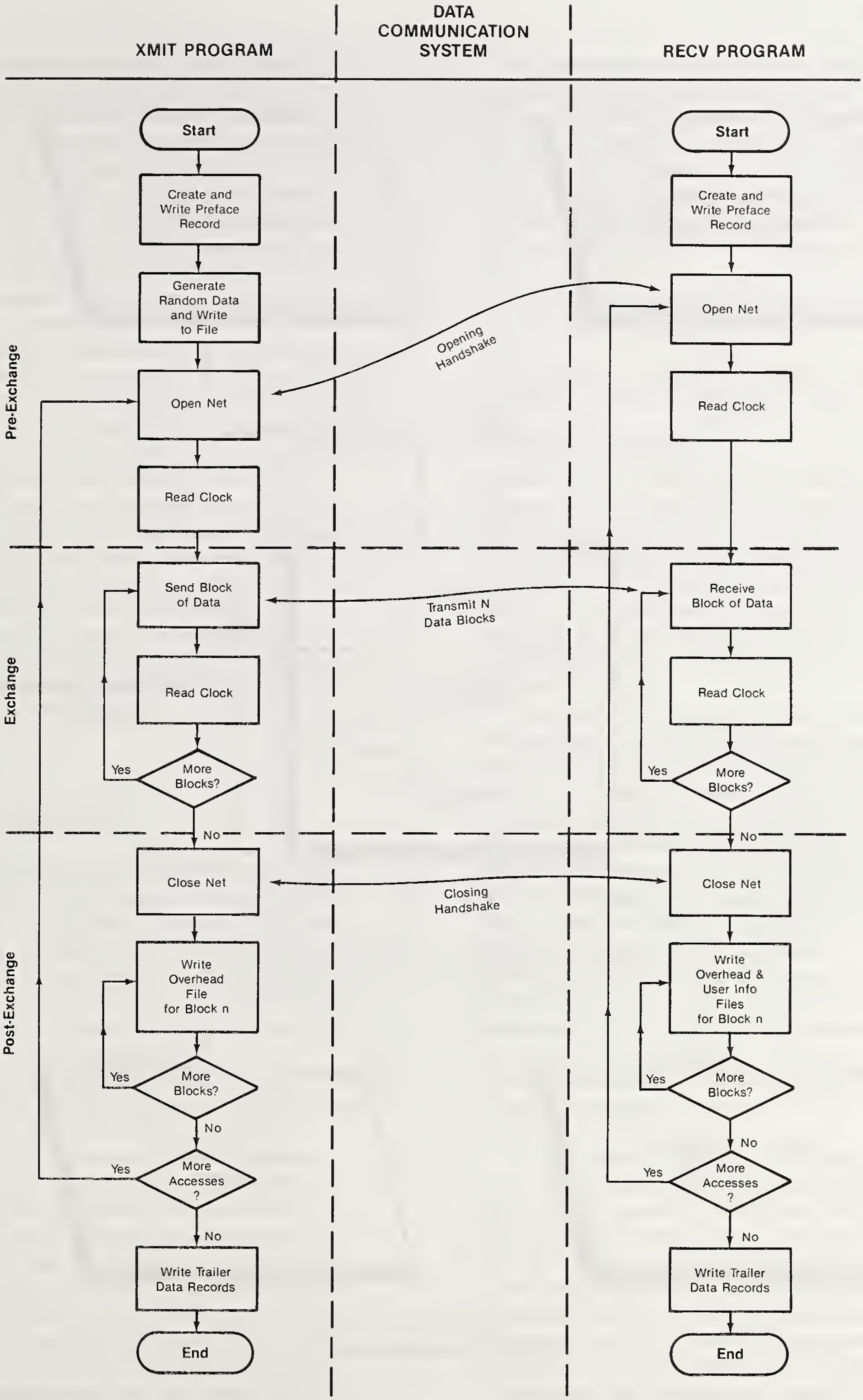


Figure B7 – Test Program Flowcharts

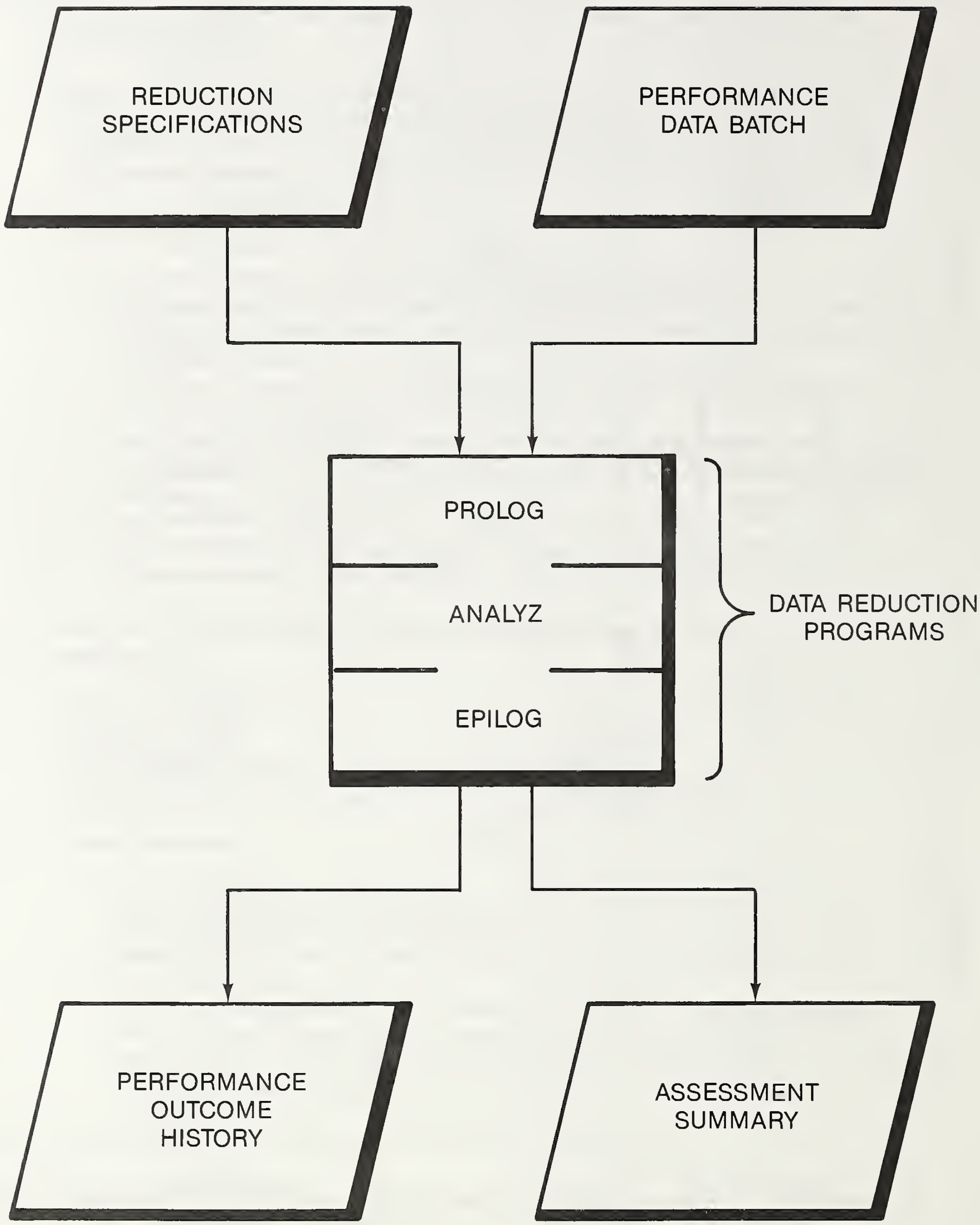


Figure B8
Data Reduction Scheme

phase of the experiment. When performance data assessment for a run is complete, ANALYZ uses the resulting performance statistics to calculate estimated values for the specified performance parameters.

EPILOG is the last program executed in a reduction run. It produces the performance assessment summary, which contains a comprehensive listing of reduction specifications, performance statistics, and measured performance parameter values.

Table B5 summarizes hypothesized results of the data reduction process for the postulated experiment. Access and disengagement results are based on an aggregate performance data sample that includes 130 of the 144 access/disengagement tests conducted during the experiment. User information transfer results are based on an aggregate sample obtained from 216 of the 240 block transfer tests. Tests excluded from the aggregate samples are postulated to have been defective. The standard deviations listed in the table are presumed to have been calculated off-line, using individual times recorded in the performance outcome files.

B4. Data Analysis

This section describes the final phase of the hypothetical experiment -- the data analysis phase. In this phase, results from individual tests are combined to calculate overall sample means and confidence intervals for each of the specified parameters. The calculated confidence intervals are then used to determine the outcome of the experiment.

For each specified parameter, the first step in the analysis is to examine the reduced data for the existence of statistically distinct groups of performance trials. If none are found, the overall parameter estimate and confidence interval are calculated from a single sample obtained by aggregating all relevant performance trials. Otherwise, the calculation requires a more elaborate analysis that takes into account the mean and standard error for each distinct group. The relevant statistical procedures are discussed in [B5]; those which apply to a single sample are implemented by the Statistical Design and Analysis computer program described in [B2].

In the postulated experiment, an exhaustive search for possibly distinct groups is not feasible. A practical alternative is to select and analyze several key groups of performance trials thought to represent the more important possible subpopulations. In the hypothetical experiment, it has been decided that the analysis should include the following examinations:

(1) Check for significant differences among the individual tests conducted on a given user pair.

(2) Check for significant differences among the tested user pairs of a given connection type.

(3) Check for significant differences between the two performance data samples obtained by combining tests for all user pairs corresponding to the Local-to-Local and Local-to-Regional connection types in one group, and tests for the remaining user pairs in another group. (In the first group, both users in a given pair are located in the same region; in the second group, the two users are in different regions.)

To simplify the application example, it is assumed that no statistically distinct groups of performance trials are found. (Such an assumption might be reasonable, for instance, in a system utilizing a communications satellite.) For each specified parameter, the relevant performance trials from all tests are therefore aggregated in a single sample. The resulting data are then used to calculate parameter estimates and the associated confidence intervals. Key performance statistics for the aggregate sample have been summarized in Table B5. Procedures and programs described in [B4] are used in checking for statistical differences and in calculating parameter estimates and the associated confidence intervals.

To determine the outcome of the experiment, the confidence interval calculated for each specified parameter is compared with the fully satisfactory range for the parameter. If, for each parameter, the calculated confidence interval intersects the fully satisfactory range, the system is considered to have passed the acceptance test. If, for one or more parameters, the calculated confidence interval fails to intersect the fully satisfactory range, the system is considered to have failed the acceptance test. Note that a system can pass

Table B5
Aggregate Sample Statistics

PRIMARY FUNCTION OUTCOME	TOTAL TRIALS	TOTAL OUTCOMES	TOTAL TRIALS LESS USER RESPONSBILE FAILURES	FAILURE PROBABILITY	AVERAGE PERFORMANCE TIME (Seconds)	SAMPLE STANDARD DEVIATION OF PERFORMANCE TIME (Seconds)	TRANSFER RATE (Bits/Second)	AVERAGE USER PERFORMANCE TIME (Seconds)	AVERAGE USER FRACTION OF PERFORMANCE TIME
ACCESS	A=1300		A′=1253						
SUCCESSFUL INCORRECT DENIAL OUTAGE USER BLOCKING		$A_s=1238$ $A_m=2$ $A_\ell=6$ $A_o=7$ $A_f=47$		 1.60×10^{-3} 4.79×10^{-3} 5.59×10^{-3}	10.62	1.84		1.80	0.169
BLOCK TRANSFER	B2=21,600		B2′=21,468						
SUCCESSFUL INCORRECT LOST EXTRA REFUSED		$B2_s=21{,}465$ $B2_e=2$ $B2_\ell=1$ $B2_x=0$ $B2_f=132$		 9.32×10^{-5} 4.66×10^{-5} 0	1.94	0.37		0.33	0.170
TRANSFER SAMPLE	B3=7128		B3′=6996*						
SUCCESSFUL DENIAL REJECTED		$B3_s=6991$ $B3_\ell=5$ $B3_f=0$		 7.15×10^{-4}	1.24	0.26	1236	0.20	0.161
DISENGAGEMENT (SOURCE)	D=1238	D′=1157							
SUCCESSFUL DENIAL USER BLOCKING		$D_s=1156$ $D_\ell=1$ $D_f=81$		 8.63×10^{-4}	2.54	0.52		0	0

*This number excludes 132 discarded transfer samples that contained Refused Blocks.

an acceptance test even through the *estimated values* for one or more specified parameters are worse than their defined thresholds, as long as the *calculated confidence interval* for each such marginal parameter extends into the fully satisfactory range for the parameter.

Table B6 summarizes the results of the hypothetical experiment. For each specified parameter, the following information is shown:

(1) The threshold value originally specified in defining the experiment objectives

(2) The decision criterion for accepting the tested hypothesis

(3) The lower 90% confidence limit

(4) The value measured in the experiment (overall sample mean)

(5) The upper 90% confidence limit

(6) A "pass" or "fail" decision based on the stated criterion

As Table B6 indicates, all specified parameters satisfy their individual acceptance criteria and therefore the overall decision is to accept the system. The calculated confidence intervals associated with the time parameter estimates clearly demonstrate that the precision objective of ±10% was exceeded by selecting, for each of the two test types, a single sample size based on the precision objective for the lowest failure probability among the set of specified parameters.

B5. References

[B1] Wortendyke, D.R.; Seitz, N.B.; Spies, K.P.; Crow, E.L.; Grubb, D.S. User-oriented performance measurements on the ARPANET: The testing of a proposed Federal Standard. Report - National Telecommunications and Information Administration, 1982; Report No. 82-112. Available from: National Technical Information Service, Springfield, VA, 22161.

[B2] Miles, M.J. Sample size and precision in communication performance measurements. Report - National Telecommunications and Information Administration, 1984; Report No 84-153. Available from: National Technical Information Service, Springfield, VA, 22161.

[B3] Seitz, N.B.; Grubb, D.S. American National Standard X3.102 user reference manual. Report - National Telecommunications and Information Administration, 1983; Report No 83-125. Available from: National Technical Information Service, Springfield, VA, 22161.

[B4] Kamas, G.; Howe, S.L. Time and frequency user's manual. Washington, DC: National Bureau of Standards, November 1979; National Bureau of Standards Special Publication No 559. Available from: Superintendent of Documents, U.S. Government Printing Office, Washington, DC, 20402.

Table B6
Completed Measurement Results Summary
(Hypothesis Test Experiment)

TESTED HYPOTHESIS: Population value of parameter lies in fully satisfactory range
SIGNIFICANCE LEVEL: 5%

PERFORMANCE PARAMETER	SPECIFIED VALUE	DECISION CRITERION	MEASUREMENT RESULTS			
			ESTIMATED VALUES			DECISION
			LOWER CONFIDENCE LIMIT	MEAN	UPPER CONFIDENCE LIMIT	
1. Access Time	12.0 s	LCL≤10.8 s	10.53 s	10.62 s	10.71 s	Pass
2. Incorrect Access Probability	1×10^{-3}	LCL≤3.16×10^{-4}	3.15×10^{-3}	1.60×10^{-3}	5.20×10^{-3}	Pass
3. Access Denial Probability	1×10^{-2}	LCL≤3.16×10^{-3}	2.16×10^{-3}	4.79×10^{-3}	9.55×10^{-3}	Pass
4. Access Outage Probability	1×10^{-2}	LCL≤3.16×10^{-3}	2.71×10^{-3}	5.59×10^{-3}	1.06×10^{-2}	Pass
5. Bit Error Probability	N/A	N/A		N/A		N/A
6. Bit Misdelivery Probability	N/A	N/A		N/A		N/A
7. Bit Loss Probability	N/A	N/A		N/A		N/A
8. Extra Bit Probability	N/A	N/A		N/A		N/A
9. Block Transfer Time	2.5 s	LCL≤2.25 s	1.94 s	1.94 s	1.94 s	Pass
10. Block Error Probability	1×10^{-4}	LCL≤3.16×10^{-5}	1.83×10^{-5}	9.32×10^{-5}	3.05×10^{-4}	Pass
11. Block Misdelivery Probability	N/A	N/A		N/A		N/A
12. Block Loss Probability	1×10^{-4}	LCL≤3.16×10^{-5}	0	4.66×10^{-5}	1.96×10^{-4}	Pass
13. Extra Block Probability	1×10^{-4}	LCL≤3.16×10^{-5}	0	0	1.94×10^{-4}	Pass
14 User Information Bit Transfer Rate	1000 bps	UCL≥1100 bps	1231 bps	1236 bps	1241 bps	Pass
15. Transfer Denial Probability	1×10^{-3}	LCL≤3.16×10^{-4}	2.92×10^{-4}	7.14×10^{-4}	1.53×10^{-3}	Pass
16. Disengagement Time	3.0 s	LCL≤2.7 s	2.51 s	2.54 s	2.57 s	Pass
17. Disengagement Denial Probability	1×10^{-3}	LCL≤3.16×10^{-4}	0	8.63×10^{-4}	3.62×10^{-3}	Pass
18. User Fraction of Access Time	0.15*	N/A		0.162*		N/A
19. User Fraction of Block Transfer Time	0.13*	N/A		0.154*		N/A
20. User Fraction of Sample Input/Output Time	0.13*	N/A		0.140*		N/A
21. User Fraction of Disengagement Time	0*	N/A		0*		N/A

N/A—Not Applicable in this example experiment
*Values included for refernce only.

X3.113-1987 Programming Language FULL BASIC
X3.114-1984 Alphanumeric Machines; Coded Character Sets for Keyboard Arrangements in ANSI X4.23-1982 and X4.22-1983
X3.115-1984 Unformatted 80 Megabyte Trident Pack for Use at 370 tpi and 6000 bpi (General, Physical, and Magnetic Characteristics)
X3.116-1986 Recorded Magnetic Tape Cartridge, 4-Track, Serial 0.250 Inch (6.30 mm) 6400 bpi (252 bpmm), Inverted Modified Frequency Modulation Encoded
X3.117-1984 Printable/Image Areas for Text and Facsimile Communication Equipment
X3.118-1984 Financial Services — Personal Identification Number — PIN Pad
X3.119-1984 Contact Start/Stop Storage Disk, 158361 Flux Transitions per Track, 8.268 Inch (210 mm) Outer Diameter and 3.937 inch (100 mm) Inner Diameter
X3.120-1984 Contact Start/Stop Storage Disk
X3.121-1984 Two-Sided, Unformatted, 8-Inch (200-mm), 48-tpi, Double-Density, Flexible Disk Cartridge for 13 262 ftpr Two-Headed Application
X3.122-1986 Computer Graphics Metafile for the Storage and Transfer of Picture Description Information
X3.124-1985 Graphical Kernel System (GKS) Functional Description
X3.124.1-1985 Graphical Kernel System (GKS) FORTRAN Binding
X3.124.2-1988 Graphical Kernel System (GKS) PASCAL Binding
X3.125-1985 Two-Sided, Double-Density, Unformatted 5.25-Inch (130-mm), 48-tpi (1,9-tpmm), Flexible Disk Cartridge for 7958 bpr Use
X3.126-1986 One- or Two-Sided Double-Density Unformatted 5.25-Inch (130-mm), 96 Tracks per Inch, Flexible Disk Cartridge
X3.127-1987 Unrecorded Magnetic Tape Cartridge for Information Interchange
X3.128-1986 Contact Start-Stop Storage Disk — 83 000 Flux Transitions per Track, 130-mm (5.118-Inch) Outer Diameter and 40-mm (1.575-Inch) Inner Diameter
X3.129-1986 Intelligent Peripheral Interface, Physical Level
X3.130-1986 Intelligent Peripheral Interface, Logical Device Specific Command Sets for Magnetic Disk Drive
X3.131-1986 Small Computer Systems Interface
X3.132-1987 Intelligent Peripheral Interface — Logical Device Generic Command Set for Optical and Magnetic Disks
X3.133-1986 Database Language — NDL
X3.135-1986 Database Language — SQL
X3.136-1988 Serial Recorded Magnetic Tape Cartridge for Information Interchange, Four and Nine Track
X3.137-1988 Unformatted Flexible Disk Cartridge, 90 mm (3.5 Inch) 5.3 tpmm (135 tpi) for 7958 bpr Use
X3.138-1988 Information Resource Dictionary System (IRDS)
X3.139-1987 Fiber Distributed Data Interface (FDDI) Token Ring Media Access Control (MAC)
X3.140-1986 Open Systems Interconnection — Connection Oriented Transport Layer Protocol Specification
X3.141-1987 Data Communication Systems and Services — Measurement Methods for User-Oriented Performance Evaluation
X3.146-1987 Device Level Interface for Streaming Cartridge and Cassette Tape Drives
X3.147-1988 Intelligent Peripheral Interface — Device Generic Command Set for Magnetic Tape Drives
X3.148-1988 Fiber Distributed Data Interface (FDDI) — Token Ring Physical Layer Protocol (PHY)
X3.153-1987 Open Systems Interconnection — Basic Connection Oriented Session Protocol Specification
X3.156-1987 Nominal 8-Inch Rigid Disk Removable Cartridge
X3.157-1987 Recorded Magnetic Tape for Information Interchange, 3200 CPI
X3.158-1987 Serial Recorded Magnetic Tape Cassette for Information Interchange, 0.150 Inch (3.81 mm), 8000 bpi (315 bpmm), Group Code Recording
X3.162-1988 Two-Sided, High-Density, Unformatted, 5.25-Inch (130-mm), 96 tpi, Flexible Disk Cartridge for 13 262 ftpr Use
X3.163-1988 Contact Start-Stop Metallic Film Storage Disk—83 333 Flux Transitions per Track, 130-mm (5.118-in) Outer Diameter and 40-mm (1.575-in) Inner Diameter
X3.165-1988 Programming Language DIBOL
X11.1-1977 Programming Language MUMPS
IEEE 416-1978 Abbreviated Test Language for All Systems (ATLAS)
IEEE 716-1982 Standard C/ATLAS Language
IEEE 717-1982 Standard C/ATLAS Syntax
IEEE 770X3.97-1983 Programming Language PASCAL
IEEE 771-1980 Guide to the Use of ATLAS
ISO 8211-1986 Specifications for a Data Descriptive File for Information Interchange
MIL-STD-1815A-1983 Reference Manual for the Ada Programming Language
NBS-ICST 1-1986 Fingerprint Identification — Data Format for Information Interchange

---

X3/TRI-82 Dictionary for Information Processing Systems (Technical Report)

American National Standards for Information Processing

X3.1-1987 Synchronous Signaling Rates for Data Transmission
X3.4-1986 Coded Character Sets — 7-Bit ASCII
X3.5-1970 Flowchart Symbols and Their Usage
X3.6-1965 Perforated Tape Code
X3.9-1978 Programming Language FORTRAN
X3.11-1969 General Purpose Paper Cards
X3.14-1983 Recorded Magnetic Tape (200 CPI, NRZI)
X3.15-1976 Bit Sequencing of the American National Standard Code for Information Interchange in Serial-by-Bit Data Transmission
X3.16-1976 Character Structure and Character Parity Sense for Serial-by-Bit Data Communication in the American National Standard Code for Information Interchange
X3.17-1981 Character Set for Optical Character Recognition (OCR-A)
X3.18-1974 One-Inch Perforated Paper Tape
X3.19-1974 Eleven-Sixteenths-Inch Perforated Paper Tape
X3.20-1967 Take-Up Reels for One-Inch Perforated Tape
X3.21-1967 Rectangular Holes in Twelve-Row Punched Cards
X3.22-1983 Recorded Magnetic Tape (800 CPI, NRZI)
X3.23-1985 Programming Language COBOL
X3.25-1976 Character Structure and Character Parity Sense for Parallel-by-Bit Data Communication in the American National Standard Code for Information Interchange
X3.26-1980 Hollerith Punched Card Code
X3.27-1987 Magnetic Tape Labels and File Structure
X3.28-1976 Procedures for the Use of the Communication Control Characters of American National Standard Code for Information Interchange in Specified Data Communication Links
X3.29-1971 Specifications for Properties of Unpunched Oiled Paper Perforator Tape
X3.30-1986 Representation for Calendar Date and Ordinal Date
X3.31-1988 Identification of the Counties of the United States
X3.32-1973 Graphic Representation of the Control Characters of American National Standard Code for Information Interchange
X3.34-1972 Interchange Rolls of Perforated Tape
X3.37-1987 Programming Language APT
X3.38-1988 Identification of States of the United States (Including the District of Columbia)
X3.39-1986 Recorded Magnetic Tape (1600 CPI, PE)
X3.40-1983 Unrecorded Magnetic Tape (9-Track 800 CPI, NRZI; 1600 CPI, PE; and 6250 CPI, GCR)
X3.41-1974 Code Extension Techniques for Use with the 7-Bit Coded Character Set of American National Standard Code for Information Interchange
X3.42-1975 Representation of Numeric Values in Character Strings
X3.43-1986 Representations of Local Time of Day
X3.44-1974 Determination of the Performance of Data Communication Systems
X3.45-1982 Character Set for Handprinting
X3.46-1974 Unrecorded Magnetic Six-Disk Pack (General, Physical, and Magnetic Characteristics)
X3.47-1988 Identification of Named Populated Places, Primary County Divisions, and Other Locational Entities of the United States
X3.48-1986 Magnetic Tape Cassettes (3.81-mm [0.150-Inch] Tape at 32 bpmm [800 bpi], PE)
X3.49-1975 Character Set for Optical Character Recognition (OCR-B)
X3.50-1986 Representations for U.S. Customary, SI, and Other Units to Be Used in Systems with Limited Character Sets
X3.51-1986 Representations of Universal Time, Local Time Differentials, and United States Time Zone References
X3.52-1976 Unrecorded Single-Disk Cartridge (Front Loading, 2200 BPI) (General, Physical, and Magnetic Requirements)
X3.53-1976 Programming Language PL/I
X3.54-1986 Recorded Magnetic Tape (6250 CPI, Group Coded Recording)
X3.55-1982 Unrecorded Magnetic Tape Cartridge, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.56-1986 Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.57-1977 Structure for Formatting Message Headings Using the American National Standard Code for Information Interchange for Data Communication Systems Control
X3.58-1977 Unrecorded Eleven-Disk Pack (General, Physical, and Magnetic Requirements)
X3.60-1978 Programming Language Minimal BASIC
X3.61-1986 Representation of Geographic Point Locations
X3.62-1987 Paper Used in Optical Character Recognition (OCR) Systems
X3.63-1981 Unrecorded Twelve-Disk Pack (100 Megabytes) (General, Physical, and Magnetic Requirements)
X3.64-1979 Additional Controls for Use with American National Standard Code for Information Interchange
X3.66-1979 Advanced Data Communication Control Procedures (ADCCP)
X3.72-1981 Parallel Recorded Magnetic Tape Cartridge, 4 Track, 0.250 Inch (6.30 mm), 1600 bpi (63 bpmm), Phase Encoded
X3.73-1980 Single-Sided Unformatted Flexible Disk Cartridge (for 6631-BPR Use)
X3.74-1981 Programming Language PL/I, General-Purpose Subset
X3.76-1981 Unformatted Single-Disk Cartridge (Top Loading, 200 tpi 4400 bpi) (General, Physical, and Magnetic Requirements)
X3.77-1980 Representation of Pocket Select Characters
X3.78-1981 Representation of Vertical Carriage Positioning Characters in Information Interchange
X3.79-1981 Determination of Performance of Data Communications Systems That Use Bit-Oriented Communication Procedures
X3.80-1988 Interface between Flexible Disk Cartridge Drives and Their Host Controllers
X3.82-1980 One-Sided Single-Density Unformatted 5.25-Inch Flexible Disk Cartridge (for 3979-BPR Use)
X3.83-1989 ISO Registration According to ISO 2375 — ANSI Sponsorship Procedures
X3.84-1981 Unformatted Twelve-Disk Pack (200 Megabytes)(General, Physical, and Magnetic Requirements
X3.85-1981 1/2-Inch Magnetic Tape Interchange Using a Self Loading Cartridge
X3.86-1980 Optical Character Recognition (OCR) Inks
X3.88-1981 Computer Program Abstracts
X3.89-1981 Unrecorded Single-Disk, Double-Density Cartridge (Front Loading, 2200 bpi, 200 tpi) (General, Physical, and Magnetic Requirements)
X3.91M-1987 Storage Module Interfaces
X3.92-1981 Data Encryption Algorithm
X3.93M-1981 OCR Character Positioning
X3.94-1985 Programming Language PANCM
X3.95-1982 Microprocessors — Hexadecimal Input/Output, Using 5-Bit and 7-Bit Teleprinters
X3.96-1983 Continuous Business Forms (Single-Part)
X3.98-1983 Text Information Interchange in Page Image Format (PIF)
X3.99-1983 Print Quality Guideline for Optical Character Recognition (OCR)
X3.100-1983 Interface Between Data Terminal Equipment and Data Circuit-Terminating Equipment for Packet Mode Operation with Packet Switched Data Communications Network
X3.101-1984 Interfaces Between Rigid Disk Drive(s) and Host(s)
X3.102-1983 Data Communication Systems and Services — User-Oriented Performance Parameters
X3.103-1983 Unrecorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 Inch)
X3.104-1983 Recorded Magnetic Tape Minicassette for Information Interchange, Coplanar 3.81 mm (0.150 in), Phase Encoded
X3.105-1983 Data Link Encryption
X3.106-1983 Modes of Operation for the Data Encryption Algorithm
X3.108-1988 Physical Layer Interface for Local Distributed Data Interfaces to a Nonbranching Coaxial Cable Bus
X3.110-1983 Videotex/Teletext Presentation Level Protocol Syntax
X3.111-1986 Optical Character Recognition (OCR) Matrix Character Sets for OCR-M
X3.112-1984 14-in (356-mm) Diameter Low-Surface-Friction Magnetic Storage Disk

(Continued on reverse)

January 1989