NTIA Report 95-319 (1)

Performance Evaluation of Data Communication Services: NTIA Implementation of American National Standard X3.141
Volume 1. Overview

Martin J. Miles

U.S. DEPARTMENT OF COMMERCE
Ronald H. Brown, Secretary

Larry Irving, Assistant Secretary
for Communications and Information

August 1995

PREFACE

Standards groups have developed methods to evaluate the performance of data communication services. Generally these methods are required to be

- user-oriented,
- system-independent, and
- uniform.

Methods to achieve these requirements are specified by two related standards approved by the American National Standards Institute (ANSI):

- ANS X3.102. The first standard (ANSI, 1983) specifies what is to be evaluated: this is a set of user-oriented and system-independent performance parameters.

- ANS X3.141. The second standard (ANSI, 1987) specifies how the ANS X3.102 performance parameters are to be evaluated: they are to be evaluated uniformly.

To support the development of these standards, two measurement programs were conducted by the Institute for Telecommunication Sciences (ITS), the research and engineering arm of the National Telecommunications and Information Administration (NTIA).

The first measurement program assessed the data communication service provided to a pair of application programs installed in two host computers on the Defense Communication Agency's Advanced Research Projects Agency Network (ARPANET): one was located at ITS in Boulder, CO, and the other was located at the National Bureau of Standards' Institute for Computer Standards and Technology (ICST) in Gaithersburg, MD (Wortendyke et al., 1982; Seitz et al., 1983).*

The second measurement program was much more extensive and had two objectives:

- Develop System. The first objective was to develop a computer-based measurement system capable of evaluating data communication services in accordance with ANS X3.102 and ANS X3.141.

- Demonstrate Ability. The second objective was to demonstrate its ability to evaluate the data communication services provided by representative public data networks and switched telephone networks.

---

*The National Bureau of Standards is now the National Institute of Standards and Technology.

An NTIA publication (Spies et al., 1988) describes the capability of the measurement system and its application to the performance evaluation of these networks.

The performance evaluation system has since been significantly enhanced and used to evaluate data communication performance in wide area networks (Wortendyke et al., 1989 and Wortendyke and Butler, 1991) and to design experiments to quantify communication satellite system performance (Cass and Miles, 1990).

The purpose of this report is to describe the NTIA implementation of the methods specified by the two standards. It consists of six volumes:

- Volume 1. This volume is an overview of the two standards and an overview of the NTIA implementation.

- Volumes 2-5. Each of these four volumes describes the NTIA implementation of one of the four specified phases: experiment design, data extraction, data reduction, and data analysis.

- Volume 6. The sixth volume explains how to graph data of the primary time performance parameters. Since ANS X3.141 recommends (but does not require) these graphical methods as part of the data analysis phase, they should be viewed as an NTIA enhancement.

This implementation (or a suitable version of it) can be used by experimenters whose objectives are acceptance, characterization, maintenance, selection, design, management, and optimization.

The following conventions are used for emphasis in this six-volume report.

- Capitalization. Names of reference events, outcomes, and performance parameters are capitalized. Also capitalized are the timeouts that terminate performance periods and Input Time and Output Time for block transfer attempts.

- Boldface. Software code is in boldface type. Code includes file names, program names, commands, input/output, etc.

- Underline. Some words, phrases, clauses, and sentences are underlined for emphasis.

- Shaded areas. Some regions are shaded for two reasons:

    - Tables. Elements in matrix-like tables are shaded to distinguish them from their defining qualities (e.g., See Table 1).

    - Actions. Actions that the experimenter must take to implement the NTIA design are contained in shaded panels (having a solid horizontal bar across both the top and the bottom).

The authors would like to thank the staff of the Center for Information Technology at the University of Wyoming for their assistance and suggestions derived from their adaptation of the NTIA implementation to the TCP/IP protocols.

CONTENTS

Page

PREFACE iii

FIGURES ix

TABLES xiii

ABSTRACT 1

INTRODUCTION 1

PART I. ANSI SPECIFICATIONS 8

1. ANS X3.102 (PERFORMANCE PARAMETERS) 9
   - 1.1 Introduction 9
   - 1.2 Data Communication Process 9
   - 1.3 Performance Parameter Definition Process 28

2. ANS X3.141 (MEASUREMENT METHODS) 55
   - 2.1 Introduction 55
   - 2.2 Experiment Design 55
   - 2.3 Data Extraction 61
   - 2.4 Data Reduction 66
   - 2.5 Data Analysis 81

PART II. NTIA DESIGN 86

3. DESIGN THE EXPERIMENT 89
   - 3.1 Introduction 89
   - 3.2 Determine the Objective of the Experiment and Select its Analysis 91
   - 3.3 Select the Most Important Performance Parameters 96
   - 3.4 Define the Population of Each Communication Function 96
   - 3.5 Specify Performance Values 97
   - 3.6 Conduct a Preliminary Characterization Test and Determine the Test Duration 97
   - 3.7 Select the Design 98
   - 3.8 Determine the Sample Size 99

4. EXTRACT THE DATA 100
   - 4.1 Introduction 100
   - 4.2 Determine the Protocol 100
   - 4.3 Determine Entity Responsibility 103
   - 4.4 Assign State Codes to Entities 103
   - 4.5 Draw the Session Profile 105
   - 4.6 Create the Protocol File 112
   - 4.7 Modify the Transmitting Program 114
   - 4.8 Create the End User Identification Files 114
   - 4.9 Work with the Ports and Create the Clock Calibration File 114

CONTENTS (Cont'd)

	Page
4.10 Set the Time Limit for Responses	115
4.11 Conduct a Test	115
4.12 Process the Test Data	115
4.13 Data Extraction Output	116
5. REDUCE THE DATA	121
5.1 Introduction	121
5.2 Synopsis of Data Reduction	121
5.3 Data Reduction Input	121
5.4 Assess Performance	123
5.5 Produce Performance Assessment Summary	144
6. ANALYZE THE DATA	152
6.1 Introduction	152
6.2 Estimate a Performance Parameter	162
6.3 Determine Acceptability of a Performance Parameter	164
6.4 Compare a Performance Parameter from Two Services	167
6.5 Determine if a Variable Condition Affects a Performance Parameter	167
7. DISPLAY THE DATA	172
7.1 Introduction	172
7.2 Draw Histograms	173
7.3 Draw Box Plots	173
7.4 Draw Chronological Plots	175
7.5 Draw Regression Plots	175
7.6 Create Box, Chronological, and Regression Plots on the HP7475A	178
PART III. NTIA CODE	180
8. DISKETTES	181
8.1 Software Diskettes	181
8.2 Demonstration Diskettes	181
9. SOFTWARE INSTALLATION	183
10. DIRECTORIES	185
10.1 Directory **/usr/net**	185
10.2 Directory **/usr/data**	187
11. REFERENCES	191
APPENDIX: BIT AND BLOCK TRANSFER OUTCOMES AS DEFINED BY ANS X3.102 . .	193
INDEX	201

FIGURES

Page

Figure 1. Venn-type diagram of structured design. 3

Figure 2. Example of a structured design diagram and the flowchart that implements it. 4

Figure 3. Structured design diagram of ANS X3.102 specifications. 10

Figure 4. Schematic diagram of elements associated with the user information transfer function 23

Figure 5. Scheme that determines access outcomes according to ANS X3.102. 33

Figure 6. Scheme that determines block transfer outcomes according to ANS X3.102 (time criteria only). 41

Figure 7. Scheme that determines disengagement outcomes according to ANS X3.102. 52

Figure 8. Structured design diagram of ANS X3.141 specifications. 56

Figure 9. Venn diagram of interface events, primary reference events, and ancillary reference events. 63

Figure 10. Flowchart that determines access outcomes according to ANS X3.141. 67

Figure 11. Venn diagram of Bit Comparison Outcomes (BCOs). 70

Figure 12. Schematic diagram of Bit Comparison Outcomes (BCOs). 71

Figure 13. Flowchart that determines bit transfer outcomes according to ANS X3.141. 72

Figure 14. Flowchart that determines block transfer outcomes according to ANS X3.141. 74

Figure 15. Flowchart that determines transfer availability trial outcomes according to ANS X3.141. 75

Figure 16. Flowchart that determines disengagement outcomes according to ANS X3.141. 76

Figure 17. Structured design diagram of procedures to determine user performance times. 78

Figure 18. Flowchart that determines user performance time outcomes according to ANS X3.141. 82

FIGURES (Cont'd)

Page

Figure 19. Structured design diagram of the NTIA design of ANS X3.141 specifications. 87

Figure 20. Structured design diagram of the NTIA design of experiment design. 92

Figure 21. Schematic diagram of a data communication system and end users. 101

Figure 22. Structured design diagram of NTIA operator implementation of data extraction. 102

Figure 23. Session profile for the schematic protocol. 107

Figure 24. Schematic protocol file. 114

Figure 25. Example of source (SOI) and destination (DOI) overhead files. 117

Figure 26. Example of source (SUI) and destination (DUI) user information files. 119

Figure 27. Flowchart that determines access outcomes according to NTIA. 125

Figure 28. Example of access outcome (ACO) file. 126

Figure 29. Flowchart that determines bit transfer outcomes and block transfer outcomes according to NTIA. 129

Figure 30. Flowchart that determines transfer availability trial outcomes according to NTIA. 132

Figure 31. Examples of bit outcome (B1O) and block outcome (B2O) files. 134

Figure 32. Examples of transfer sample outcome (B3O) and throughput sample outcome (B4O) files. 136

Figure 33. Flowchart that determines disengagement outcomes according to NTIA. 141

Figure 34. Examples of source disengagement outcome (D1O) and destination disengagement outcome (D2O) files. 142

Figure 35. Example of access assessment summary file. 145

FIGURES (Cont'd)

Page

Figure 36. (Part 1). Example of user information transfer assessment summary file. 146

Figure 36. (Part 2). Example of user information transfer assessment summary file. 147

Figure 36. (Part 3). Example of user information transfer assessment summary file. 148

Figure 37. Example of source disengagement assessment summary file. . . 150

Figure 38. Example of destination disengagement assessment summary file. 151

Figure 39. Example from shell script implementation of analysis of failure probability performance parameters from an access-disengagement test. 154

Figure 40. Structured design diagram of operator implementation of analysis of single tests. 156

Figure 41. Flowchart that determines pooling for analysis of multiple tests. 158

Figure 42. Example of analysis of Access Time and User Fraction of Access Time from multiple tests. 159

Figure 43. Schematic operating characteristic curve of a sampling plan for an acceptance test. 166

Figure 44. Example of a test of variable conditions for a time performance parameter. 171

Figure 45. Example of histogram of Source Disengagement Time. 174

Figure 46. Example of box plots of Block Transfer Times. 176

Figure 47. Example of chronological plots of Block Transfer Times. . . . 177

Figure 48. Example of regression plots of Block Transfer Times. 179

Figure 49. Directory structure of NTIA code. 186

Figure 50. (Part 1). List of files in each subdirectory of **usr/net**. . . 188

Figure 50. (Part 2). List of files in each subdirectory of **usr/net**. . . 188

Figure 50. (Part 3). List of files in each subdirectory of **usr/net**. . . 188

FIGURES (Cont'd)

Page

Figure A-1. Block transfer attempt containing Extra Bit outcomes. 196

TABLES

Page

Table 1. User-System Interfaces Depending upon the End User and the Data Media Condition 13

Table 2. Examples of User-System Interface Events Depending upon the End User and the Data Media Condition 14

Table 3. Primary Reference Events 20

Table 4. ANS X3.102 Performance Parameters 31

Table 5. Bit Transfer Outcomes (Except Misdelivered Bit Outcome) and Their Defining Conditions 39

Table 6. Block Transfer Outcomes (Except Misdelivered Block Outcome) and Their Defining Conditions 43

Table 7. Transfer Availability Trial Outcomes and Their Defining Conditions 48

Table 8. Combinations of Responsibilities for the Three Entity-Interface Pairs 65

Table 9. Relevant Interfaces for Performance Time Allocation 80

Table 10. Decisions Required to Design an Experiment of Data Communication Services 90

Table 11. Common Experiment Objectives and Plausible Analyses 96

Table 12. Commands and Expected Responses and Their Corresponding Primary Reference Events 104

Table 13. Bit Transfer Outcomes (Except Misdelivered Bit Outcome) Depending upon Bit Comparison Outcomes and Block Transfer Outcomes 130

Table 14. Bit and Block Transfer Outcomes That Can Coexist 130

Table 15. Available Plots for Primary Delay Performance Parameters 173

Table A-1. Bit Transfer Outcomes as a Function of Timeliness-Entity Responsibility, Accuracy, and Transfer 198

Table A-2. Block Transfer Outcomes as a Function of Timeliness-Entity Responsibility, Accuracy, and Start of Block Transfer 200

**PERFORMANCE EVALUATION OF DATA COMMUNICATION SERVICES:
NTIA IMPLEMENTATION OF AMERICAN NATIONAL STANDARD X3.141**

VOLUME 1. OVERVIEW

Martin J. Miles[1]

The six volumes of this report are:

Volume 1. Overview
Volume 2. Experiment Design
Volume 3. Data Extraction
Volume 4. Data Reduction
Volume 5. Data Analysis
Volume 6. Data Display

This volume is an overview of the American National Standards Institute (ANSI) specifications for performance evaluation of data communication services and the NTIA implementation. The specifications are described in ANS X3.102 (the standard that specifies the user-oriented and system-independent performance parameters) and ANS X3.141 (the standard that specifies uniform methods to evaluate them). The performance parameters are evaluated in four phases: experiment design, data extraction, data reduction, and data analysis. A fifth phase, data display, is an NTIA enhancement.

Key words: acceptance tests; access; American National Standards; analysis; analysis of variance (ANOVA); availability; box plots; characterization; data communication functions; data communication services; data communication session; dependability; dependence; disengagement; estimation; experiment design; extraction; factors; graphics; histograms; hypothesis tests; interface events; linear regression; performance parameters; precision; protocol; reduction; reference events; sample size; session profile; speed; structured design; user information transfer

INTRODUCTION

Most large projects, including large software projects, are planned and described by a method called structured design. Structured design is usually considered to have four increasingly specific phases: requirements,

[1]The author is with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, CO 80303.

considered to have four increasingly specific phases: requirements, specifications, design, and code.[2] Figure 1 is a Venn-type diagram that illustrates the increasing specificity of the four phases. (Errors occurring in the less specific phases are more costly to correct.) The phases of structured design can be summarized as follows:

- Requirements. The requirements phase usually indicates only what is to be done. It usually does not indicate how, when, where, or who is to do it; those designations are deferred to more specific phases. In some applications the requirements are so general that they are either omitted or joined with the specifications phase.

- Specifications. The specifications phase is less general than the requirements phase. It also indicates what is to be done, but with some qualifications. These qualifications usually specify values to be achieved.

- Design. The design phase begins the implementation of the specifications. This phase shows how the requirements and specifications are to be accomplished. The design can be described by two types of diagrams - structured design diagrams and flowcharts:

    - Structured Design Diagrams. These diagrams depict what is to be done. They have three components: Input/output that are described by words within rectangles, processes that are described by words within ovals (called modules), and arrows that connect input/output and processes and indicate the direction of input/output or the process. A module can represent one or more processes. Those that represent more than one process, can be divided into modules that represent single processes when the design is better understood. Figure 2a is an example of a simple structured design diagram that indicates that the function F is to be evaluated for values of the independent variable, x.

    - Flowcharts. These diagrams are constructed after the structured design diagrams are completed. They show how each process (module) of a structured design diagram is implemented. They often include decisions which are described by

---

[2]The most specific phase is called "code" because structured design methods were developed for software projects.

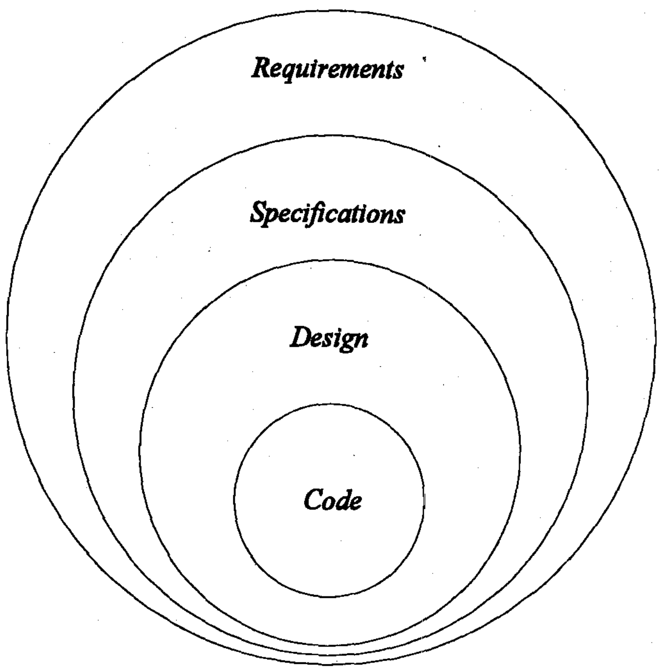
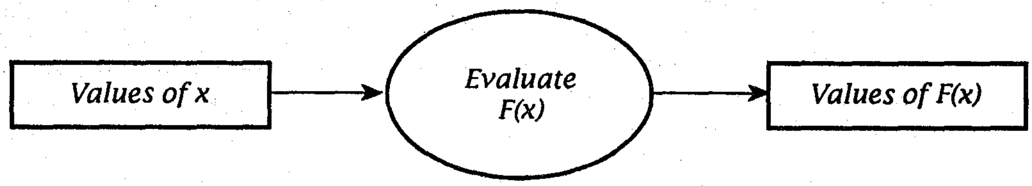
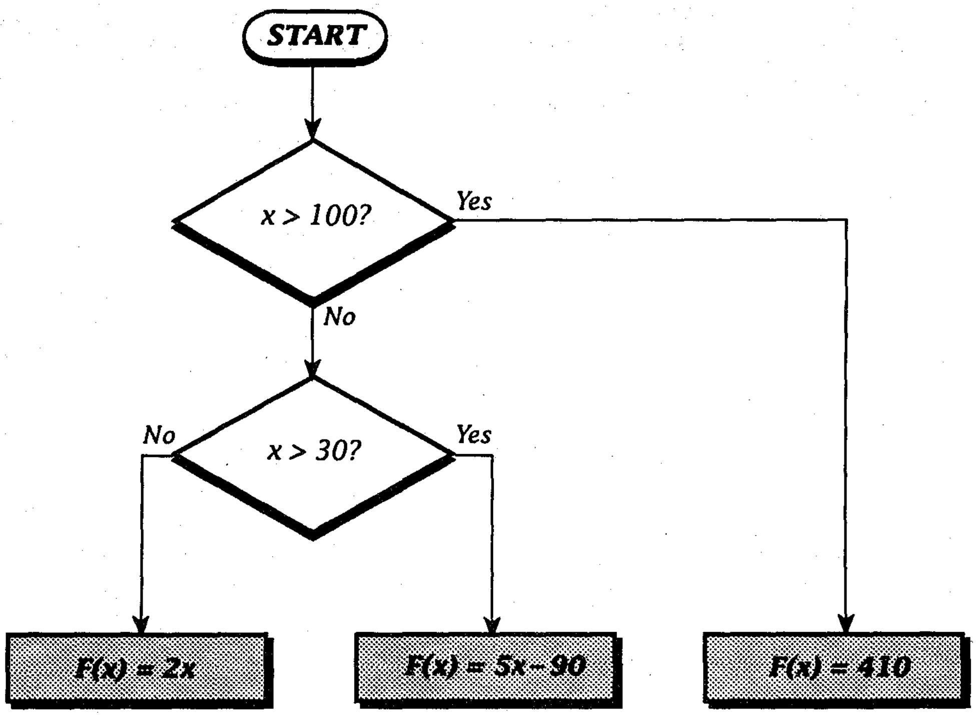


Figure 1. Venn-type diagram of structured design.



a. Structured Design Diagram.



b. Flowchart.

Figure 2. Example of a structured design diagram and the flowchart that implements it.

illustrates how the structured design diagram of Figure 2a is to be implemented.

- Code. The code phase implements the design. It is the most specific phase (i.e., the product). For a book, code would be the text; for a device, code would be the assembled components; and for software, code would be the characters comprising the instructions.

Structured design shows how the requirements, specifications, design and code are applied to evaluate services provided by the data communications industry. Specifically,

- structured design provides an excellent framework for a system to evaluate the performance of data communication services (the subject of this report), and

- structured design provides an excellent framework to evaluate the performance of a *particular* data communication service.[3]

The following two panels illustrate these two applications.

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

SYSTEM TO EVALUATE THE PERFORMANCE OF DATA COMMUNICATION SERVICES

- Requirements. The performance of data communication services is to be evaluated in a way that is
  - user-oriented,
  - system-independent, and
  - uniform.
- Specifications. The specifications are provided by two related ANSI standards:
  - ANS X3.102 specifies what is to be evaluated - a set of user-oriented and system-independent performance parameters, and
  - ANS X3.141 specifies how they are to be evaluated - uniformly.

  Part I of this volume is an overview of ANS X3.102 and ANS X3.141.
- Design. A design to implement these specifications is provided by NTIA (the subject of this report). Part II of this volume is an overview of the NTIA design of the four phases (experiment design, data extraction, data reduction, and data analysis), and a fifth phase, data display.
- Code. Code consists of software and documentation - both provided by NTIA. Software is included in seven diskettes. Documentation consists of this six volume report and comments imbedded in the software. Part III of this volume is an overview of the NTIA software.

EVALUATE THE PERFORMANCE OF A PARTICULAR DATA COMMUNICATION SERVICE

Suppose the objective is to characterize some performance parameters of a data communication network that provides service among 21 end users, 24 hours a day, and Monday through Friday. The service uses the TCP/IP protocol.

- Requirements. Characterize Access Time, Throughput, and Block Error Probability for this service at user-system interfaces.
    - The performance parameters are to be defined by ANS X3.102.
    - The performance parameters are to be characterized according to ANS X3.141.
- Specifications. Some precision and budget considerations must be specified:
    - Precision specifications:
        - Access Time is to be estimated within 0.5 s of its true value at the 95% confidence level.
        - Throughput is to be estimated within 10 Kbps of its true value at the 95% confidence level.
        - Block Error Probability is to be estimated within 50% of its true value at the 90% confidence level.
    - Budget specifications:
        - The experiment is to be conducted over five representative pairs of end users, four times a day, and five days a week. Two representative block sizes are to be used.
        - The design of the experiment is to be a randomized complete block design.
- Design and Code. The requirements and specifications are to be implemented by the design and code provided by NTIA, but the code for the data extraction phase must be modified to match the TCP/IP protocol.

In accordance with structured design, this volume is organized in three parts. Part I is an overview of the ANSI specifications, Part II is an overview of the NTIA design, and Part III is an overview of the NTIA software code.

PART I. ANSI SPECIFICATIONS

The methods to evaluate the performance of data communication services are required to be

- user-oriented,
- system-independent, and
- uniform.

These requirements are met by methods specified in two related ANSI standards:

- ANS X3.102. The first standard (ANSI, 1983) specifies what is to be evaluated: These are a set of user-oriented and system-independent performance parameters.
- ANS X3.141. The second standard (ANSI, 1987) specifies how the ANS X3.102 performance parameters are to be evaluated: They are to be evaluated uniformly.

Part I is an overview of these two standards. It is not intended to either reproduce or replace them.

Note that the topics in each section of Part I agree with those in the similar section in each of the two standards.

PART I. ANS X3.102 SPECIFICATIONS

1. ANS X3.102 (PERFORMANCE PARAMETERS)

1.1 Introduction

ANS X3.102-1983 was revised in 1992. Although the NTIA implementation has not been completely upgraded to reflect the revised version, <u>this section is an overview of the revised version</u> because this version is an improvement, and (with the exception of some user information transfer performance parameters) the performance parameters are defined in the same way.

This standard defines 24 user-oriented and system-independent performance parameters that can be used to evaluate the performance of data communication services. The performance parameters define performance from the end user's point-of-view, and they are system-dependent because they are defined in terms of events that have the same purpose. Definition of the performance parameters requires two processes: The first process (Section 1.2) is the data communication process, and the second process (Section 1.3) is the performance parameter definition process. Figure 3 is a structured design diagram describing these two processes. Each process is described by a shaded oval; however, the performance parameter definition process is separated into two - one for access/disengagement performance parameters and one for user information transfer performance parameters. Performance parameters are the output from these processes.

1.2 Data Communication Process

The data communication process involves seven major concepts. They are

1. end users,
2. data communication system,
3. user-system interfaces,
4. interface events,
5. transferred information,
6. data communication session, and
7. primary data communication functions.

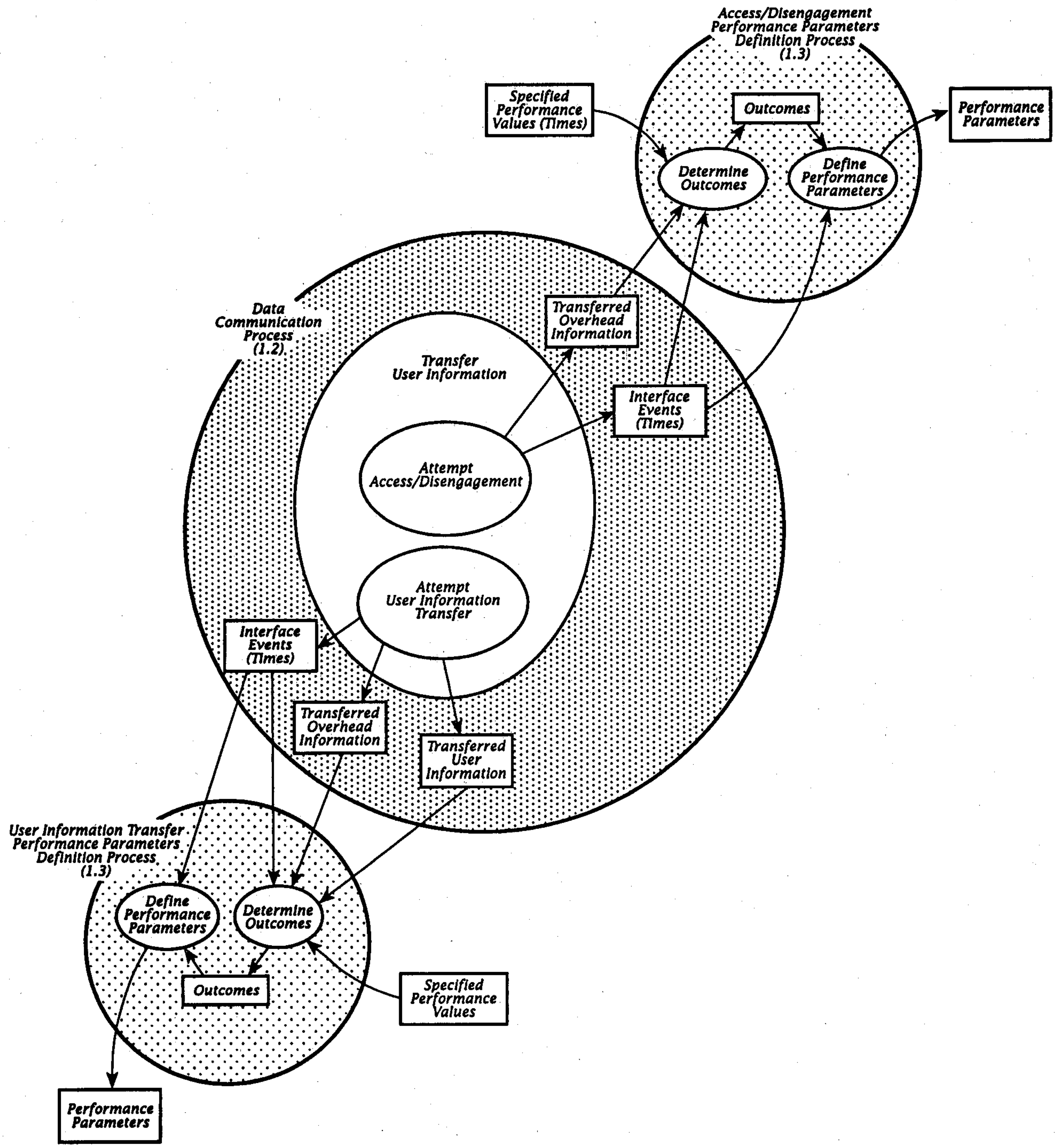


Figure 3. Structured design diagram of ANS X3.102 specifications.

The first two concepts are entities in the data communication process, and the third concept exists between the entities. The remaining four concepts are depicted in the structured design diagram (Figure 3); since the fourth and fifth concepts are input/output, they are described in rectangles, and since the sixth and seventh concepts are processes, they are described in ovals. The process labelled "Transfer User Information" is accomplished by the data communication session. The data communication session is comprised of three primary data communication functions. The subprocess labelled "Attempt Access/Disengagement" is accomplished by the two primary data communication functions, access and disengagement, and the subprocess labelled "Attempt User Information Transfer" is accomplished by the primary data communication function, user information transfer.

1.2.1 End Users

An end user is an entity that may be an operator of an input/output device or an application program that simulates an operator.

End users can be divided into four types, depending upon their participation. They can be source or destination end users, and/or they can be originating or nonoriginating end users:

- Source/Destination End Users. The source end user is the end user from which user information is to be transferred, and the destination end user is the end user to which user information is to be transferred.

- Originating/Nonoriginating End Users. The originating end user is the end user that initiates the data communication session, and the nonoriginating end user is the end user that does not initiate the session.

The source end user is not necessarily the originating end user.

1.2.2 Data Communication System

A data communication system is an entity that includes all functional and physical elements that participate in transferring information between two or more end users. It consists of facilities such as transmission lines and relays (for transfer), switches (for routing), data terminals (for input/output), protocols (rules for transferring information), and operating systems (for managing the interfaces between the system and application programs).

1.2.3 User-System Interfaces

An interface is the boundary between two entities. The two entities at each interface in a data communication session are generally the end user and the local operating system (OS) of the data communication system.[4] The entities can create four types of user-system interfaces, depending upon whether the end user is a (human) operator or an application program and whether there are data media or not:

- Operator with no Data Media. If the end user is an operator with no associated data media, the user-system interface is the physical interface between the operator and the input/output device (i.e., keyboard, printer, monitor, etc.).
- Operator with Data Media. If the end user is an operator with associated data media, the user-system interface is both
    - the physical interface between the operator and the input/output device, and
    - the physical interface between the data medium and the input/output device.
- Application Program with no Data Media. If the end user is an application program with no associated (separate) data media, the user-system interface is the functional interface between the application program and the local operating system.[5]
- Application Program with Data Medium. If the end user is an application program with associated (separate) data medium, the user-system interface is both
    - the functional interface between the application program and the local operating system, and

[4]Subsystem interfaces also exist, but this implementation is intended to evaluate performance between user-system interfaces.

[5]The NTIA implementation uses an application program as the source end user (called **xmit_net**) and an application program as the destination end user (called **recv**).

- the physical interface between the medium and the input/output device.

Table 1 summarizes the user-system interfaces that depend upon the end user and whether there are data media or not.

Table 1. User-System Interfaces Depending upon the End User and the Data Media Condition

	DATA MEDIA CONDITION	
END USER	No Data Media	Data Media
Operator	•Operator - I/O Device (Physical Interface)	•Operator - I/O Device (Physical Interface) and •Data Medium - I/O Device (Physical Interface)
Application Program	•App. Program - Local OS (Functional Interface)	•App. Program - Local OS (Functional Interface) and •Data Medium - I/O Device (Physical Interface)

If a user-system interface doesn't coincide with an organizational interface [i.e., an interface created from vendor-supplied equipment, such as the interface between data terminal equipment (DTE) and data circuit-terminating equipment (DCE)], it may be desirable to use organizational interfaces if they are digital.

If networks from two vendors are used in the data communication system, it may be appropriate to consider each network to be a subsystem.

1.2.4 Interface Events

An interface event is an occurrence that causes information to cross a user-system interface. The time at which an interface event is said to cross the interface depends upon the direction of transfer of information across the interface:

- <u>User to System</u>. Information has been transferred from a user to a system when it is in the system's receiving facility, and the system has been authorized to either send it (in the case of user information) or process it (in the case of overhead information).[6]

- System to User. Information has been transferred from a system to a user when it is in the user's receiving facility, and the user has been notified that it is available.

Table 2 lists examples of interface events at the four types of user-system interfaces summarized in Table 1.

Table 2. Examples of User-System Interface Events Depending upon the End User and the Data Media Condition

	DATA MEDIA CONDITION	
END USER	No Data Media	Data Media
Operator	•Keying Characters •Printing Characters •Displaying Characters	•Reading/Writing Magnetic Media
Application Program	•Issuing System Calls •Setting/Clearing Flags[7]	•Reading/Writing Magnetic Media

1.2.5 Transferred Information

There are two types of transferred information, user information and overhead (residual) information.

A. User Information

User information is information transferred from a source user to the system that is intended to cross the destination user-system interface. User information may cross the source user-system interface in a different code than it crosses the destination user-system interface (e.g., ASCII characters may

[6] User information is the "message" the source user wishes to convey to the destination user; it is formally defined in Section 1.2.5. Overhead information is information that is not user information.

[7] A flag is a character that synchronizes bytes and frames.

change to a pattern of colors on a monitor, or depression of the space bar may change to a movement of the curser on a monitor). If the user information code at the source user-system interface is intended to differ from that at the destination user-system interface, the user information bits are defined differently for the source and the destination user:

- Source User Information Bits. Source user information bits are bits used for the binary representation of user information transferred from the source user to the system. For example, when user information is input in nonbinary form, such as by ASCII characters, source user information bits are the bits used to initially code this information.

- Destination User Information Bits. Destination user information bits are bits used for binary representation of user information transferred from the system to a destination user. For example, when information is output in nonbinary form, such as by ASCII characters, destination user information bits are the bits used to finally decode this information.

B. Overhead Information

Overhead information is information that is not user information. There are three types of overhead information:

- System Control Information. This is information transferred from a user to the system to control the system. Examples are ASCII characters such as **DLE**, **ESC**, and **ENQ**.[8]

- Intra-System Coordination Information. This is information transferred between elements of the system to coordinate their operation. This information doesn't cross user-system interfaces. Examples are parity bits (bits added for error control), ASCII characters (such as **SYN**, **ACK**, and **NAK**), flags (characters that synchronize bytes and frames), addresses (data structures that define destination users), control characters (characters or bit-patterns containing control information), control fields (fields in frames containing control information), and frame check sequence (FCS) fields of LAPB frames (defined in CCITT Recommendation X.25 for Link Access Procedure on the ISDN B channel).

---

[8]Although not stated in the standard, the most important example of system control overhead information, in the case of a user that is an application program, is the set of instructions that comprises the application program.

- User Control Information. This is information transferred from the system to a user to report status or to control the user. An example is a call progress signal in public data networks.[9]

1.2.6 Data Communication Session

A data communication session is a coordinated sequence of end user and system activities intended to transfer user information from one or more source users to one or more destination users. A normal data communication session between a pair of end users comprises a single access function, a single user information transfer function and disengagement function for each end user.

The type and sequence of interface events in the session are both system-dependent and application-dependent. The session always

- begins with the interface event that commits the originating user to attempt to transfer user information (i.e., the interface event called Access Request),
- includes all subsequent interface events that depend upon the committing interface event (above) to attempt to transfer user information, and
- ends with the interface event that renders all subsequent interface events independent of the committing interface event (i.e., the interface event called either Source Disengagement Confirmation or Destination Disengagement Confirmation).

Data communication sessions can vary fundamentally, depending upon the type of connection, the direction of user information transfer, and the number of end users.

A. Type of Connection

There are two types of connections (also called types of sessions), and they differ in two fundamental ways. They differ in the order in which user information is transferred relative to the nonoriginating user's commitment to

---

[9]A call progress signal is a two-digit number (defined by CCITT recommendation X.96) that is sent by the DCE to the DTE to inform it of the result of the call.

the session (i.e., connection establishment) and also in the order in which message elements arrive (i.e., message order):

- Connection-Oriented Session (similar to the telephone system).

    - Connection Establishment. Connection is established before transmission and disconnected after transmission. Because a connection is established, user information must be input to the system after the nonoriginating user has committed to participate in the data communication session.

    - Message Order. All message elements (e.g., words, sentences, etc.) arrive in order, because they use the same path.

  Circuit-switching and virtual circuit packet-switching are examples of a connection-oriented session because a (dedicated) path is established between source and destination users for the duration of the session.

- Connectionless Session (similar to the postal system).

    - Connection Establishment. Connection is not established before transmission. Because connection need not be established before transmission, user information may be input to the system before the nonoriginating user has committed to participate in the data communication session.

    - Message Order. Message elements (e.g., letters) generally arrive in the order sent - but not necessarily, because they can use different paths.[10]

  Packet-switching is an example of a connectionless session because a path is not established for all packets; they can be transmitted along independent paths between the source and destination user.

The distinction between the two types of connections is necessary to determine access outcomes as discussed in Section 1.3.1.

---

[10]A datagram is an example of a packet or a short message that is transmitted without a connection having been established, and the packets are transmitted independently.

B. Directions of User Information Transfer and Number of End Users

Sessions can involve either one-way (simplex) or two-way (duplex) transmission, and there may be multiple pairs of end users.[11] A data communication session can be described by a separate session profile for each pair of end users and for each direction of user information transfer.

1.2.7 Primary Data Communication Functions

The primary data communication functions in a data communication session are defined in terms of a single pair of end users. They are the access function, the user information transfer function, and the disengagement function.

To define the primary data communication functions and their performance parameters in a way that is system-independent, it is necessary to identify those interface events from any data communication system that have the same purpose. Those that do are given the same name. For example, the interface event that initiates a data communication session and commits the originating user to participate probably has a name that depends upon the system. However, all

---

[11]The NTIA implementation is designed to collect and process performance data from sessions involving one-way (i.e., simplex) transmission between a single pair of users. If the direction of transfer is two-way (i.e., duplex) and/or the number of pairs is greater than one, the on-line data extraction software must be modified:

- Each session is treated as the superposition of two simplex sessions.
- A given user acts as the source in one session and as the destination in the other.
- Each interface monitor generates one set of extracted data files in which the local user acts as the source and another set in which the local user acts as the destination.
- These sets of extracted data are input to separate data conversion runs to produce two sets of performance data - one for each direction of transfer.
- The two sets of extracted data are processed (i.e., reduced and analyzed) in the usual manner.

interface events that do this (i.e., initiate and commit) can be (and are) given the name, Access Request. For example, in X.25 services, a **Call Request** packet (that is placed in the calling DTE output queue) is an Access Request.

Interface events that define primary data communication functions are called primary reference events. (Ancillary reference events also exist; they are discussed in Section 2.3.2.) Table 3 is a list of the three primary data communication functions, the ten primary reference events, and a sentence describing the effect of each primary reference event upon the data communication session.

Table 3. Primary Reference Events

PRIMARY DATA COMMUNICATION FUNCTIONS	PRIMARY REFERENCE EVENTS	DATA COMMUNICATION SESSION EFFECTS
Access	Access Request	Requests initiation of a data communication session and commits the originating user to participate.
	Nonoriginating User Commitment	Indicates intent of specified nonoriginating (called) user to participate in a requested data communication session.
	System Blocking Signal	Notifies the originating user that the system will not support a requested data communication session.
	User Blocking Signal	Notifies the system that the issuing user will not support a requested data communication session.
	Access Confirmation	Indicates that the source user is able to initiate user information transfer after connection establishment (in a connection-oriented data communication session)
User Information Transfer	Start of Block Transfer	Authorizes the system to transfer a single source user information block.
	End of Block Transfer	Completes transfer to destination user of destination block that contains last destination bit associated with a block transfer attempt.
Disengagement	Disengagement Request	Requests termination of a user's participation in an established data communication session.
	User Disengagement Blocking Signal	Indicates intent of issuing user to continue data communication session to transfer additional user information.
	Disengagement Confirmation	Indicates termination of a user's participation in an established data communication session.

The primary data communication functions described here are normal functions (i.e., they are completed without failure); nonnormal functions (called exceptional functions) end in failure and are discussed in Section 1.3, where 14 of the 24 performance parameters are seen to be failure probability performance parameters.

A. Access Function

The access function comprises those activities that the end users and the system must accomplish in order for the system to accept user information (for transfer to the destination user).

This primary function spans the following two primary reference events:

- Access Request. The access function begins with the Access Request reference event or its implied equivalent at the interface between the originating user and the system. The Access Request reference event may be issued by either a user or the system.

- Access Confirmation. The access function normally ends with the Access Confirmation reference event. However, when no such signal is provided by the system, the Access Confirmation reference event occurs implicitly when the source user is able to initiate user information transfer (after the Nonoriginating User Commitment reference event in connection-oriented sessions).

Three additional primary reference events may occur during the access function:

- Nonoriginating User Commitment. This reference event indicates the intent of the called user to participate in the requested data communication session. This reference event is necessary in a connection-oriented session.

- System Blocking Signal. This reference event notifies the originating user that the system will not support the data communication session.

- User Blocking Signal. This reference event notifies the system that the issuing end user (either end user) will not support the data communication session.

B. User Information Transfer Function

The user information transfer function comprises those activities that the end users and system must accomplish in order to transfer user information from the source user through the system to the destination user.

This primary function spans the following two primary reference events:

- Start of Block Transfer. The user information transfer function begins with the Start of Block Transfer reference event for the first source user information block - a time that corresponds with the end of the access function (i.e., the Access Confirmation reference event).

- End of Block Transfer. The user information transfer function ends with the End of Block Transfer reference event for the last destination user information block. (This event occurs when the last bit of destination user information is output from the system to the destination user.)

The standard defines two user information transfer functions, one for blocks and one for bits (a necessary distinction because some systems accept blocks having different numbers of bits). The standard also defines a transfer sample to be a number of consecutive block transfer attempts and their preceding interblock gaps. However, it does not define a transfer function for these sequences. For consistency of presentation, this overview defines the transfer of these sequences to be a function also.

Thirteen performance parameters are associated with the user information transfer function; they are distributed as follows:

- Blocks. Six performance parameters are associated with the block transfer function.

- Bits. Four performance parameters are associated with the bit transfer function.

- Transfer Samples. Three performance parameters are associated with transfer samples - whose transfers will be called (in this overview only) the transfer sample transfer function.

Figure 4 shows some elements necessary to define the user information transfer function; the three entities (i.e., source user, system, and destination user) and their two interfaces are shown. Time flows down along the interfaces during the session, and block transfer attempts are represented by thick lines.

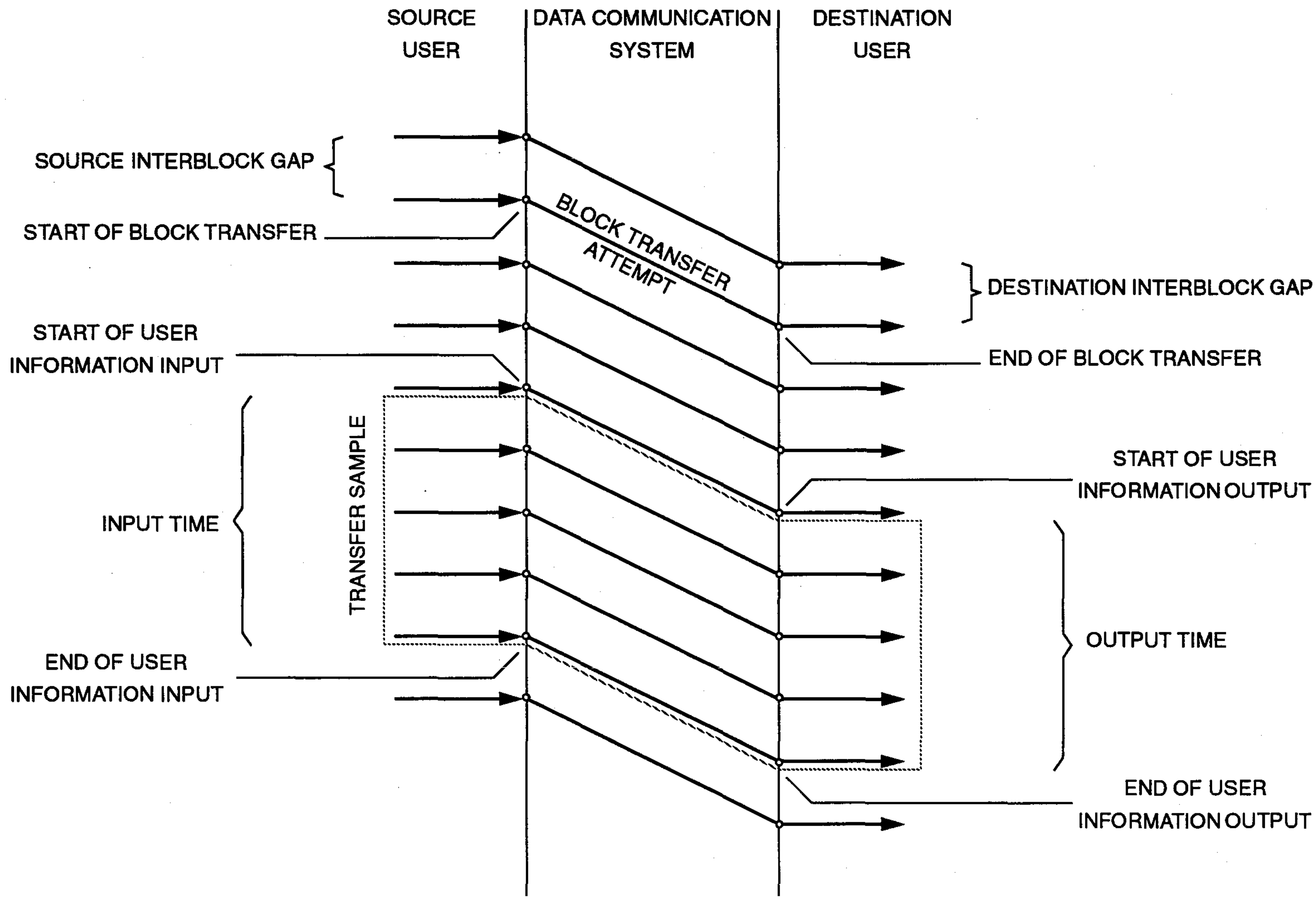


Figure 4. Schematic diagram of elements associated with the user information transfer function.

1. Block Transfer Function

A user information block consists of a number of consecutive bits that may define an ASCII character, a computer word, a data packet, or the user information field of a frame - depending upon the equipment and protocol associated with the user-system interfaces. Typically, user information that is input to the system at the source user-system interface is partitioned into a sequence of source user information blocks, and user information that is output by the system at the destination user-system interface is partitioned into a sequence of destination user information blocks.

The block transfer function comprises those user and system activities necessary to transfer a block of source user information through the system to the destination user. The block transfer function for each source user information block and its corresponding destination user information block spans the following two primary reference events:

- Start of Block Transfer. The block transfer function begins, for any source user information block, when the block has been input to the system, and the system has been authorized to begin its transfer.

- End of Block Transfer. The block transfer function ends when the destination user information block that contains the last user information bit, corresponding to a bit in the transferred source user information block, is output from the system (with appropriate notification to the destination user, if required).

2. Bit Transfer Function

Typically, the sizes of source user information blocks and destination information blocks correspond. However, since they do not always correspond, the standard provides a necessarily precise description of user information transfer performance that distinguishes bit transfer and block transfer.[12]

[12]Systems can be composed of a packet-switching system at the source user-system interface that employs packets of a certain size and a packet-switching system at the destination user-system interface that employs packets of a different size.

The bit transfer function is associated with the transfer of a bit of source user information through the system to the destination user. The bit transfer function for each source user information block spans the following two events (which correspond to primary reference events):

- Start of Bit Transfer. The bit transfer function begins, for any bit in a source user information block, when the user information block transfer function begins for that block (i.e., the Start of Block Transfer reference event).

- End of Bit Transfer. The bit transfer function ends, for any bit in a source user information block, when the user information block transfer function ends for that block (i.e., the End of Block Transfer reference event).

3. Transfer Sample Transfer Function

A transfer sample is a number of consecutive block transfer attempts and their preceding interblock gaps.[13] Hence, the transfer sample transfer function can begin with the Start of Block Transfer for the second user information block and continue through the user information transfer function. Each transfer sample spans a period of time called the Input/Output Time. For each transfer sample, the Input/Output Time is the larger of the Input Time and the Output Time. Their end points are defined as follows:

- Start of Input Time. The Input Time for a transfer sample begins with the Start of Block Transfer reference event for the last block transfer attempt preceding the transfer sample.

- End of Input Time. The Input Time for a transfer sample ends with the Start of Block Transfer reference event for the last block transfer attempt in the transfer sample.

[13]The standard defines a transfer sample as containing a number of consecutive block transfer attempts rather than a number of consecutive blocks because the blocks may contain fragments of source blocks and destination blocks. Moreover, the standard defines a transfer sample as beginning with an interblock gap only because, if it were defined as ending with an interblock gap, that gap would affect the block following the transfer sample. The duration of interblock gaps can vary due to protocol, equipment, and users (e.g., with flow control).

- Start of Output Time. The Output Time for a transfer sample begins with the End of Block Transfer reference event for the last block transfer attempt preceding the transfer sample.

- End of Output Time. The Output Time for a transfer sample ends with the End of Block Transfer reference event for the last block transfer attempt in the transfer sample.

Suppose n block transfer attempts result in a transfer sample containing n - 1 block transfer attempts and their preceding interblock gaps. A transfer sample is defined such that this transfer sample could result as many as

$$n(n - 1)/2$$

transfer samples (e.g., many could be nested and some could be concatenated).

To calculate a performance parameter that measures dependability, the experimenter must specify a minimum Input/Output Time.[14] Then a transfer availability trial is the smallest transfer sample that spans the specified Input/Output Time. Similarly, there could be many transfer availability trials - many could be nested and some could be concatenated.

To calculate performance parameters that measure the long-term throughput, a series of one or more transfer samples can be called a throughput trial.

In summary, the transfer sample transfer function uses the following concepts to define its three performance parameters:

- Transfer Sample. A transfer sample is a number of consecutive block transfer attempts and their preceding interblock gaps. Each transfer sample has an associated Input/Output Time. There may be zero, one, or many transfer samples in a transfer sample.

- Transfer Availability Trial. A transfer availability trial is the smallest transfer sample whose associated Input/Output Time exceeds the specified minimum Input/Output Time. There may be zero, one, or many transfer availability trials in a transfer sample.

- Throughput Trial. A throughput trial is a series of one or more transfer samples. There may be zero or one throughput trials.

---

[14]ANS X3.102-1983 states that the transfer sample must include a specified number of bits rather than span a specified duration.

C. Disengagement Function

The disengagement function is really two functions, one for each end user.

1. Source Disengagement Function

The source disengagement function comprises those activities that the end users and the system must accomplish to terminate the session for the source end user.

This primary function spans the following two primary reference events:

- Disengagement Request. The source disengagement function begins upon the issuance by any participating entity (i.e., the system or either end user) of the Disengagement Request reference event.

- Disengagement Confirmation. The source disengagement function ends when the Disengagement Confirmation reference event occurs. The system usually issues this reference event. However, if it does not, it occurs implicitly when a request for reestablishment of the terminated data communication session can be initiated.

This function may include the User Disengagement Blocking Signal primary reference event, but it does not include a "System Disengagement Blocking Signal" primary reference event (as does the access function) because the system doesn't possess the means to block source disengagement:

- User Disengagement Blocking Signal. This reference event indicates the intent of the issuing user (either end user) to continue the data communication session.

2. Destination Disengagement Function

The destination disengagement function comprises those activities that the end users and the system must accomplish to terminate the session for the destination end user.

This primary function spans the following two primary reference events:

- Disengagement Request. The destination disengagement function begins upon the issuance by any participating entity (i.e., the system or either end user) of the Disengagement Request reference event.

- Disengagement Confirmation. The destination disengagement function ends when the Disengagement Confirmation reference event occurs. The system usually issues this reference event. However, if it does not, it occurs implicitly when a request for reestablishment of the terminated data communication session can be initiated.

This function may include the User Disengagement Blocking Signal primary reference event, but it does not include a "System Disengagement Blocking Signal" primary reference event (as does the access function) because the system doesn't possess the means to block destination disengagement:

- User Disengagement Blocking Signal. This reference event indicates the intent of the issuing user (either end user) to continue the data communication session.

1.3 Performance Parameter Definition Process

Figure 3 is a structured design diagram that shows the data communication process (Section 1.2) and the performance parameter definition process.[15] The performance parameter definition process requires two subprocesses, one to determine both the times and the outcomes of attempts (to perform a data communication function) and one to use the times and the outcomes to define the performance parameters.

The time of each attempt is determined by the difference between two selected interface events.

The outcome of each attempt is determined by one or more of three conditions. These conditions are

- the occurrence of and relative order of interface events (and the cause for their relative order),

[15]The structured design diagram shows one performance parameter definition process (i.e., lightly shaded oval) for each type of data communication function - the access/disengagement functions and the user information transfer function. Actually, the user information transfer function is divided into three functions - one for bits, one for blocks, and one for transfer samples (in this overview), and the disengagement function is divided into two functions - one for the source end user and one for the destination end user.

- the occurrence of and relative order of interface events and the "timeout" (three times as long as a performance time specified by the experimenter), and, where appropriate,

- the quality of transferred user information.[16]

Performance parameters are defined by the difference in times of occurrences of interface events and/or the outcomes of attempts to perform a data communication function. They are divided into two types, primary performance parameters and ancillary performance parameters:

- Primary Performance Parameters. Nineteen primary performance parameters measure speed, accuracy, and dependability.

    - Speed. There are five primary speed performance parameters. Four speed performance parameters measure the delay between selected primary reference events (or their implied equivalent events). The (total) delay is the sum of the user delay and the system delay. One speed performance parameter measures the rate of user information transfer.

    - Accuracy and Dependability. Accuracy and dependability performance parameters are characterized by failures. Failures occur when the system

        - causes a primary speed performance parameter to exceed, by a factor of three, its specified performance time, resulting in a "timeout", or

        - causes bits, blocks, and transfer availability trials to be unsuccessfully transferred.

---

[16]A nonperformance time is a time after which the time to perform is so excessive that it must be considered a failure to perform. If user information is transferred from both end users (i.e., duplex communication), nonperformance times should be specified for each end user because the nature of their participation can be quite different (e.g., one user might easily request large files while the other user is burdened with writing them). Analysis of performance parameters from a session involving more than one source or destination end user can be performed in a number of ways, such as by computing the average of performance times from the participating users.

- Ancillary Performance Parameters. Five ancillary performance parameters measure the degree to which the five primary speed performance parameter are influenced by the user. Ancillary performance parameters are defined as the ratio of user delay-to-total delay:

  (user delay) / (user delay + system delay).

  The user delay is the difference in times of occurrence of selected ancillary reference events, and the total delay is the difference in times of occurrence of selected primary reference events.[17]

The speed performance parameters are measured by the total delay. The system delay for a primary delay performance parameter can be evaluated as

(primary delay performance parameter) x [1 - (ancillary delay performance parameter)].

The system rate for the primary rate performance parameter can be evaluated as

(primary rate performance parameter) / [1 - (ancillary rate performance parameter)].

Table 4 lists the 24 ANS X3.102 performance parameters. Primary performance parameters are shown in bold, and ancillary performance parameters are shown in italics. Table 4a lists the 24 ANS X3.102 performance parameters according to the primary data communication functions and according to the criteria that concern most users: speed, accuracy, and dependability. Table 4b lists the same performance parameters according to the primary data communication functions, but they are mapped according to types of random variables. (A random variable is a concept from probability and statistics that must be employed when applying experiment design and analysis, as specified in ANS X3.141.)

The following sections are organized according to the data communication functions. Within each section, it is shown how to determine outcomes of attempts to perform the function, then each performance parameter is defined in a box. Since we are interested in system performance, in all cases, an attempt to perform a data communication function whose outcome is User Blocking is omitted from performance evaluation.

---

[17]The definition of an ancillary reference event is deferred to the overview of ANS X3.141 (Section 3.3.2).

Table 4. ANS X3.102 Performance Parameters

a. Organization by primary communication function and performance criterion

PRIMARY COMMUNICATION FUNCTIONS			PERFORMANCE CRITERIA: SPEED	ACCURACY	DEPENDABILITY
		ACCESS	• ACCESS TIME • *USER FRACTION OF ACCESS TIME*	• INCORRECT ACCESS PROBABILITY	• ACCESS DENIAL PROBABILITY • ACCESS OUTAGE PROBABILITY
USER INFORMATION TRANSFER		BIT TRANSFER	—	• BIT ERROR PROBABILITY • BIT MISDELIVERY PROBABILITY • EXTRA BIT PROBABILITY	• BIT LOSS PROBABILITY
USER INFORMATION TRANSFER		BLOCK TRANSFER	• BLOCK TRANSFER TIME • *USER FRACTION OF BLOCK TRANSFER TIME*	• BLOCK ERROR PROBABILITY • BLOCK MISDELIVERY PROBABILITY • EXTRA BLOCK PROBABILITY	• BLOCK LOSS PROBABILITY
USER INFORMATION TRANSFER	TRANSFER SAMPLE TRANSFER	TRANSFER AVAIL-ABLILTY	—	• TRANSFER DENIAL PROBABILITY	
USER INFORMATION TRANSFER	TRANSFER SAMPLE TRANSFER	THROUGHPUT	• USER INFORMATION BIT TRANSFER RATE • *USER FRACTION OF INPUT/OUTPUT TIME*	—	—
DISENGAGEMENT		SOURCE DISENGAGEMENT	• SOURCE DISENGAGEMENT TIME • *USER FRACTION OF SOURCE DISENGAGEMENT TIME*	• SOURCE DISENGAGEMENT DENIAL PROBABILITY	
DISENGAGEMENT		DESTINATION DISENGAGEMENT	• DESTINATION DISENGAGEMENT TIME • *USER FRACTION OF DESTIN-ATION DISENGAGEMENT TIME*	• DESTINATION DISENGAGEMENT DENIAL PROBABILITY	

b. Organization by primary communication function and random variable

PRIMARY COMMUNICATION FUNCTIONS			RANDOM VARIABLES: DELAY	RATE	FAILURE
		ACCESS	• ACCESS TIME	• *USER FRACTION OF ACCESS TIME*	• INCORRECT ACCESS • ACCESS OUTAGE • ACCESS DENIAL
USER INFORMATION TRANSFER		BIT TRANSFER	—	—	• BIT ERROR • BIT MISDELIVERY • EXTRA BIT • BIT LOSS
USER INFORMATION TRANSFER		BLOCK TRANSFER	• BLOCK TRANSFER TIME	• *USER FRACTION OF BLOCK TRANSFER TIME*	• BLOCK ERROR • BLOCK MISDELIVERY • EXTRA BLOCK • BLOCK LOSS
USER INFORMATION TRANSFER	TRANSFER SAMPLE TRANSFER	TRANSFER AVAIL-ABLILTY	—	—	• TRANSFER DENIAL
USER INFORMATION TRANSFER	TRANSFER SAMPLE TRANSFER	THROUGHPUT	—	• USER INFORMATION BIT TRANSFER RATE • *USER FRACTION OF INPUT/OUTPUT TIME*	—
DISENGAGEMENT		SOURCE DISENGAGEMENT	• SOURCE DISENGAGEMENT TIME	• *USER FRACTION OF SOURCE DISENGAGEMENT TIME*	• SOURCE DISENGAGEMENT DENIAL
DISENGAGEMENT		DESTINATION DISENGAGEMENT	• DESTINATION DISENGAGEMENT TIME	• *USER FRACTION OF DESTINATION DISENGAGEMENT TIME*	• DESTINATION DISENGAGEMENT DENIAL

1.3.1 Access Function

A. Determine Access Outcomes

All access attempts begin with the Access Request reference event. After the Access Request reference event, access outcomes are defined by the relative times of occurrences of interface events and the Access Timeout. Figure 5 illustrates the time criteria for access outcomes.[18] An access attempt can result in one of five possible outcomes:

- Successful Access outcome. There are two types of sessions, and this outcome depends upon the type of session:

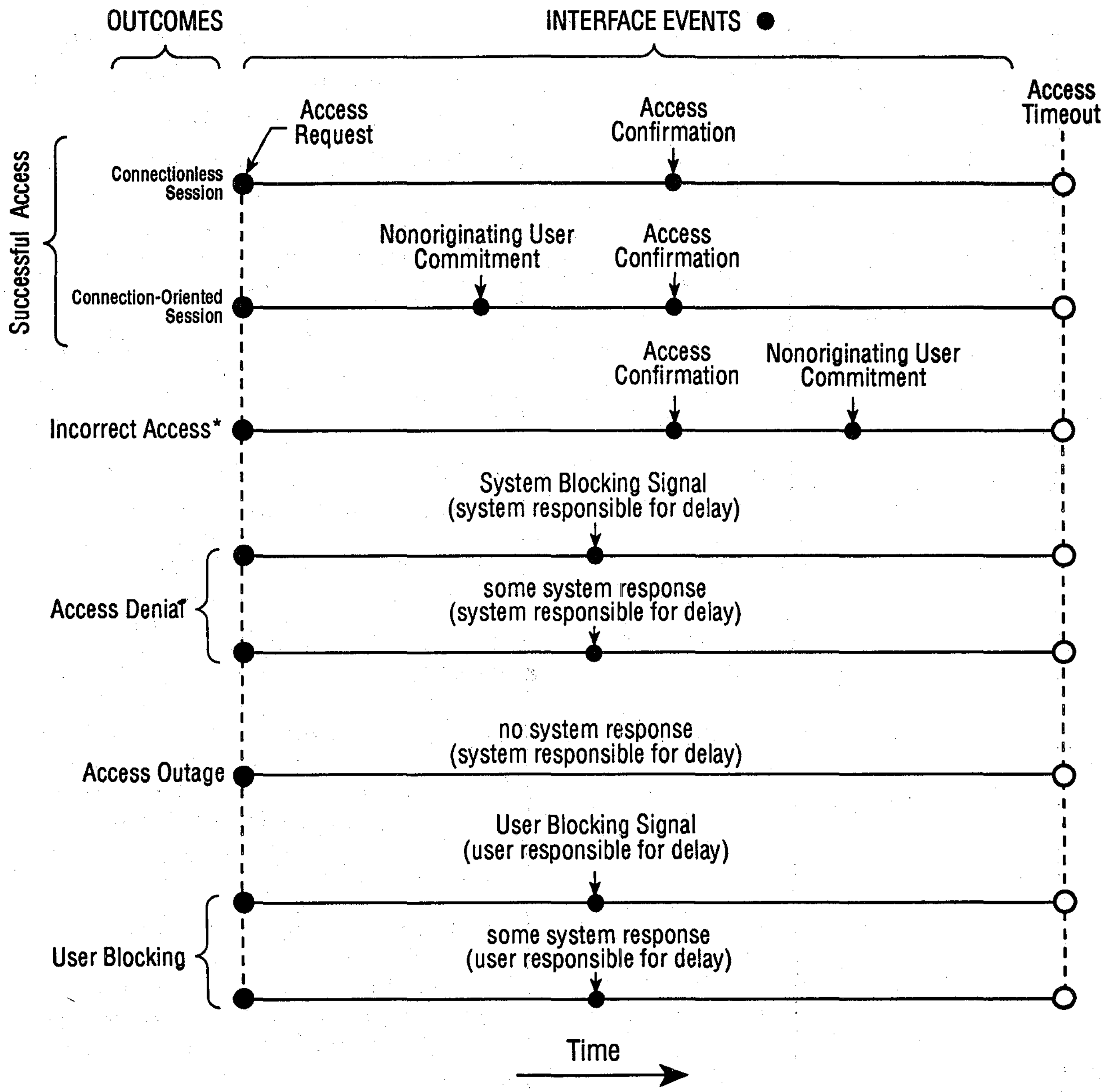
    - Connectionless Session. Successful Access outcome occurs if the Access Confirmation reference event occurs on or before Access Timeout.

    - Connection-oriented Session. Successful Access outcome occurs if the Access Confirmation reference event occurs before Access Timeout and the Nonoriginating User Commitment reference event occurs on or before the Access Confirmation reference event. (This second criterion implies that the nonoriginating user had committed to the session prior to the access attempt.)

- Incorrect Access outcome (i.e., a wrong number). This outcome occurs if the Access Confirmation reference event occurs before Access Timeout, and the Nonoriginating User Commitment reference event does not occur before the Access Confirmation reference event (in fact, it probably will not occur). This outcome occurs when the system establishes a (physical or virtual) connection to an unintended called user and does not correct the error prior to the Access Confirmation reference event. Incorrect Access outcome is distinguished from Successful Access outcome by the fact that the Incorrect Access outcome requires the Access Confirmation reference event not be preceded by the Nonoriginating User Commitment reference event (i.e., the specified nonoriginating user had not committed to the session during the access attempt).

---

[18]Reference events that might occur after Access Timeout are not shown because they are irrelevant to the outcomes.



*Incorrect Access outcome occurs only in connection-oriented sessions

Figure 5. Scheme that determines access outcomes according to ANS X3.102.

- <u>Access Denial outcome</u>. This outcome can occur if

    - a System Blocking Signal reference event is issued to the originating user before Access Timeout, or

    - some response is issued by the system to the originating user before Access Timeout, and the system is responsible for the delay.

    The Access Confirmation reference event may occur after the Access Timeout, but its occurrence would be irrelevant to the outcome.

- <u>Access Outage outcome</u>. This outcome occurs if Access Timeout occurs before the system issues a response (e.g., the system is "dead"); this outcome contrasts with the Access Denial outcome (in which the system does issue a response).

- <u>User Blocking outcome</u>. This outcome can occur if

    - an end user issues a User Blocking Signal reference event on or before the Access Timeout, or

    - the system issues a response on or before the Access Timeout, and the user is responsible for the delay.

    The Access Confirmation reference event may occur after the Access Timeout, but its occurrence would be irrelevant to the outcome.

B. Define Access Performance Parameters

Time performance parameters can be calculated from a single access attempt whose outcome is Successful Access (although it is usual to calculate them as an average of several Successful Access outcomes). Failure probability performance parameters are defined as ratios of the number of failure outcomes to the number of attempts. For the access function, the standard defines failure probability performance parameters as the ratio of the number of failure outcomes to the "total access attempts included in an access parameter determination." This total is the sum of all access attempts except those attempts whose outcome is User Blocking; for convenience, the total is referred to here as the "number of access attempts."

Five performance parameters are associated with the access function: one primary time performance parameter, its ancillary performance parameter, and three failure probability performance parameters.

1. Access Time

Access Time is calculated from Successful Access outcomes only.

Access Time = average [(time of Access Confirmation) - (time of Access Request)]

where

- Access Request is a primary reference event, and
- Access Confirmation is a primary reference event.

2. User Fraction of Access Time

User Fraction of Access Time is calculated from Successful Access outcomes only.

User Fraction of Access Time = (average User-Responsible Access Time) / (Access Time)

where

- User-Responsible Access Time is that portion of Access Time for which the user is responsible, and
- Access Time has just been defined.

3. Incorrect Access Probability

This outcome occurs only in connection-oriented sessions because the system does not establish a connection in connectionless systems.

Incorrect Access Probability = (number of Incorrect Access outcomes) / (number of access attempts)

4. Access Denial Probability

Access Denial Probability = (number of Access Denial outcomes) / (number of access attempts)

5. Access Outage Probability

Access Outage Probability = (number of Access Outage outcomes) / (number of access attempts)

1.3.2 User Information Transfer Function

User information transfer outcomes are determined by the relative time of occurrences of reference events and the Block Transfer Timeout as well as by the quality and quantity of the transferred user information. The user information transfer function can conveniently be described by dividing it into three functions: bit transfer, block transfer, and transfer sample transfer. The functions are not independent, however, because some bit transfer outcomes are defined in terms of block transfer reference events, etc.[19]

A. Bit Transfer Function

1. Determine Bit Transfer Outcomes

It is necessary to associate destination user information bits with their block transfer attempts because

- some systems do not maintain correspondence between source and destination blocks, and
- the destination user information may contain extra user information bits.

A bit transfer attempt can result in one of six possible outcomes:

- <u>Successful Bit Transfer outcome</u>. This outcome occurs if

[19]The definitions of bit transfer outcomes and block transfer outcomes from ANS X3.102-1992 are quite lengthy, so they are relegated to the Appendix. The definitions in this section reflect the intent of that standard but are abbreviated for an overview.

- a source bit is transmitted,
- a corresponding destination bit is received,
- the transmitted and received bits have the same binary values, and
- the bit transfer attempt is contained in a block transfer attempt in which the End of Block Transfer reference event occurs no later than the Block Transfer Timeout.

- <u>Incorrect Bit outcome</u>. This outcome occurs if
  - a source bit is transmitted,
  - a corresponding destination bit is received,
  - the transmitted and received bit have different binary values, and
  - the bit transfer attempt is contained in a block transfer attempt in which the End of Block Transfer reference event occurs no later than the Block Transfer Timeout.

- <u>Extra Bit outcome</u>. This outcome occurs if
  - a destination bit is received,
  - a corresponding source bit is not transmitted, and
  - the bit transfer attempt is not contained in a block transfer attempt whose outcome is Refused Block.

- <u>Misdelivered Bit outcome</u>.[20] This outcome occurs if
  - the bit transfer attempt is associated with an extra user information bit (at an unintended destination),

---

[20]When Bit Misdelivery Probability is not evaluated (as it is not in the NTIA implementation), the bit transfer attempt that would result in a Misdelivered Bit outcome is an Extra Bit outcome.

- the destination user information bit associated with the bit transfer attempt corresponds to a bit input to the system by the source user for transfer to an unintended destination user, and

- the bit transfer attempt is not included in a block transfer attempt whose outcome is Refused Block.

- <u>Lost Bit outcome</u>. This outcome occurs if

  - a source bit is transmitted,

  - either a corresponding destination bit is not received or the bit transfer attempt is contained in a block transfer attempt in which the End of Block Transfer reference event occurs later than the Block Transfer Timeout, and

  - responsibility for the excessive delay is attributed to the system.

- <u>Refused Bit outcome</u>. This outcome occurs if the bit transfer attempt is contained in a block transfer attempt whose outcome is Refused Block.

Table 5 categorizes all bit transfer outcomes except the Misdelivered Bit outcome. (Since Bit Misdelivery Probability is an optional performance parameter which is not implemented by NTIA, its inclusion in the table would complicate it unnecessarily.) The outcomes are determined by the disposition of the bits and by the disposition of the block:

- <u>Disposition of Bits</u>. The bit transfer outcomes depend upon the accuracy and the end user location of the bits (classifications that are formalized in ANS X3.141 and called data correlation).[21]

- <u>Disposition of Blocks</u>. The bit transfer outcomes depend upon the timeliness of the block transfer attempt and the entity responsible for the delay when it exceeds Block Transfer Timeout.

---

[21]Bit Location means that the bit is in the information record of the stated end user.

Table 5. Bit Transfer Outcomes (Except Misdelivered Bit Outcome) and Their Defining Conditions

BIT CORRELATION		BLOCK TRANSFER TIMELINESS-ENTITY RESPONSIBILITY		
Bit Location	Bit Accuracy	Timely	Late-User Responsible	Late-System Responsible
Source Only	--	Lost Bit	Refused Bit	Lost Bit
Destination Only	--	Extra Bit	Refused Bit	Extra Bit
Source and Destination	Same Binary Value	Successful Bit Transfer	Refused Bit	Lost Bit
	Different Binary Value	Incorrect Bit	Refused Bit	Lost Bit

2. Define Bit Transfer Performance Parameters

Four performance parameters are associated with bit transfer outcomes, and all are failure probability performance parameters.

a. Bit Error Probability

Bit Error Probability = (number of Incorrect Bit outcomes) / [(number Successful Bit Transfer outcomes) + (number of Incorrect Bit outcomes)]

b. Extra Bit Probability

Extra Bit Probability = (number of Extra Bit outcomes) / [(number of bit transfer attempts) - (number of Lost Bit outcomes)]

The "number of bit transfer attempts" paraphrases the "total bit transfer attempts that are included in a bit transfer parameter determination" as stated in the standard.

c. Bit Misdelivery Probability

This performance parameter is optional and has not been implemented by NTIA.

Bit Misdelivery Probability = (number of Misdelivered Bit outcomes) / [(number of Successful Bit Transfer outcomes) + (number of Incorrect Bit outcomes) + (number of Misdelivered Bit outcomes)]

d. Bit Loss Probability

Bit Loss Probability = (number of Lost Bit outcomes) / [(number of Successful Bit Transfer outcomes) + (number of Incorrect Bit outcomes) + (number of Lost Bit outcomes)]

B. Block Transfer Function

1. Determine Block Transfer Outcomes

Figure 6 illustrates the time criteria for block transfer outcomes. A block transfer attempt can result in one of six possible outcomes:

- Successful Block Transfer outcome. This outcome occurs if
    - the Start of Block Transfer reference event occurs,
    - the End of Block Transfer reference event occurs on or before the Block Transfer Timeout, and
    - the received user information is identical to the transferred user information.
- Incorrect Block outcome. This outcome occurs if
    - the Start of Block Transfer reference event occurs,
    - the End of Block Transfer reference event occurs no later than the Block Transfer Timeout, and
    - the received user information is not identical to the transferred user information.
- Extra Block outcome. This outcome occurs if
    - the Start of Block Transfer reference event does not occur, and
    - the End of Block Transfer reference event occurs.

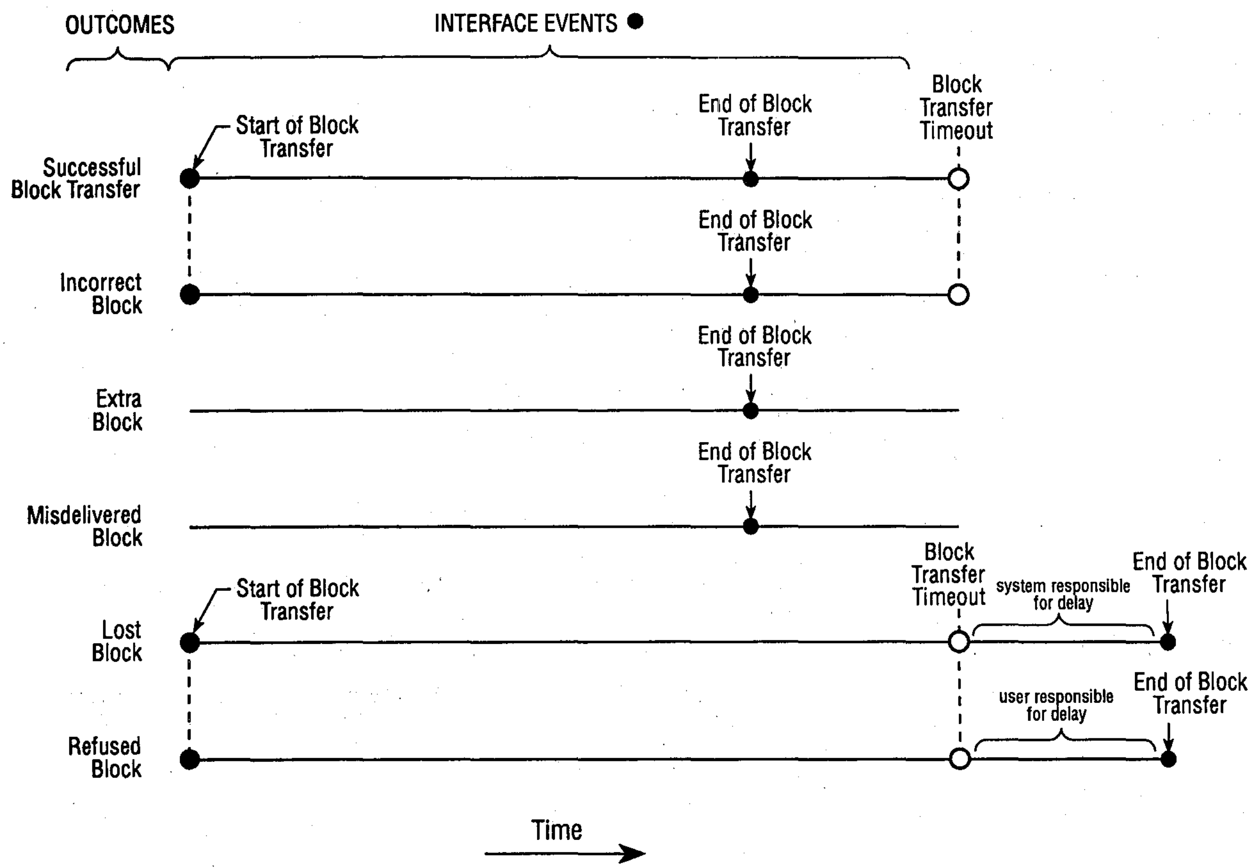


Figure 6. Scheme that determines block transfer outcomes according to ANS X3.102 (time criteria only).

- Misdelivered Block outcome.[22] This outcome occurs if
    - the block transfer attempt is associated with a destination block consisting of extra user information bits, and
    - one or more destination user information bits associated with the block transfer attempt correspond to user information bits input to the system by the source user for transfer to some other destination user.
- Lost Block outcome. This outcome occurs if
    - the Start of Block Transfer reference event occurs,
    - the End of Block Transfer reference event occurs later than the Block Transfer Timeout, and
    - responsibility for the excessive delay is attributed to the system.
- Refused Block outcome. This outcome occurs if
    - the Start of Block Transfer reference event occurs,
    - the End of Block Transfer reference event occurs later than the Block Transfer Timeout, and
    - responsibility for the excessive delay is attributed to the user.

Table 6 categorizes all block transfer outcomes except the Misdelivered Block outcome. (Since Block Misdelivery Probability is an optional performance parameter which is not implemented by NTIA, its inclusion in the table would complicate it unnecessarily.) Of course, the table contains no entry corresponding to the impossible outcome of a timely block transfer attempt in which the block exists at the source only.

[22]When Block Misdelivery Probability is not evaluated, the block transfer attempt that would result in a Misdelivered Block outcome is an Extra Block outcome.

Table 6. Block Transfer Outcomes (Except Misdelivered Block Outcome) and Their Defining Conditions

BLOCK CORRELATION		BLOCK TRANSFER TIMELINESS-ENTITY RESPONSIBILITY		
Block Location	Bit Accuracy	Timely	Late-User Responsible	Late-System Responsible
Source Only	--	--	Refused Block	Lost Block
Destination Only	--	Extra Block	Refused Block	Extra Block
Source and Destination	All Same Binary Value	Successful Block Transfer	Refused Block	Lost Block
	Some Different Binary Value	Incorrect Block	Refused Block	Lost Block

2. Define Block Transfer Performance Parameters

Six performance parameters are associated with these outcomes: One primary time performance parameter, its ancillary performance parameter, and four failure probability performance parameters. Time performance parameters can be calculated from a single block transfer attempt whose outcome is Successful Block Transfer (although it is usual to report them as an average of several Successful Block Transfer outcomes).

a. Block Transfer Time

This performance parameter is calculated from Successful Block Transfer outcomes only.

> Block Transfer Time = average [(time of End of Block Transfer) - (time of Start of Block Transfer)]

where

- End of Block Transfer is a primary reference event, and
- Start of Block Transfer is a primary reference event.

This performance parameter may be used to determine the "response time" of a remote-access data processing service.

b. User Fraction of Block Transfer Time

This performance parameter is calculated from Successful Block Transfer outcomes only.

User Fraction of Block Transfer Time = (average User-Responsible Block Transfer Time) / (Block Transfer Time)

where

- User-Responsible Block Transfer Time is that portion of Block Transfer Time for which the user is responsible, and
- Block Transfer Time has just been defined.

The time required for the user to load paper into a printer is an example of the User-Responsible Block Transfer Time.

c. Block Error Probability

Block Error Probability = (number of Incorrect Block outcomes) / [(number of Successful Block Transfer outcomes) + (number of Incorrect Block outcomes)]

d. Extra Block Probability

Extra Block Probability = (number of Extra Block outcomes) / [(number of block transfer attempts) - (number of Lost Block outcomes)]

The "number of block transfer attempts" paraphrases the "total block transfer attempts included in a block transfer parameter determination" as stated in the standard.

e. Block Misdelivery Probability

This performance parameter is optional and has not been implemented by NTIA.

Block Misdelivery Probability = (number of Misdelivered Block outcomes) / [(number of Successful Block Transfer outcomes) + (number of Incorrect Block outcomes) + (number of Misdelivered Block outcomes)]

f. Block Loss Probability

Block Loss Probability = (number of Lost Block outcomes) / [(number of Successful Block Transfer outcomes) + (number of Incorrect Block outcomes) + (number of Lost Block outcomes)]

C. Transfer Sample Transfer Function

A transfer sample is a number of consecutive block transfer attempts and their preceding interblock gaps. The number and size of transfer samples can vary considerably.[23]

A transfer availability trial is the smallest transfer sample (i.e., the smallest transfer sample whose Input/Output Time exceeds the specified minimum Input/Output Time).

1. Determine Transfer Sample Transfer Outcomes

The standard does not define outcomes for transfer samples, but it does define outcomes for transfer availability trials. Transfer availability trial outcomes are determined from four bit-oriented user information transfer

[23]Because reference events are required to define Input Times and Output Times, both the block transfer attempt preceding the transfer sample and the block transfer attempt ending the transfer sample must have outcomes that are either Successful Block Transfer or Incorrect Block.

performance parameters that are called supported performance parameters.[24] They are

- Bit Error Probability,
- Bit Loss Probability,
- Extra Bit Probability, and
- User Information Bit Transfer Rate (UIBTR).

A threshold value is defined for each supported performance parameter. It is a function of the specified performance value. Specifically, the threshold of each supported failure probability performance parameter is the square root of its specified performance value, and the threshold of the User Information Bit Transfer Rate is one-third of its specified performance value. (For purposes of a supported performance parameter, the UIBTR is also defined for a single transfer sample.)

A transfer availability trial can result in one of three outcomes:

- <u>Successful Transfer outcome</u>. This outcome occurs if the observed value of a transfer availability trial is equal to or better than the threshold of <u>each</u> of the four supported performance parameters.
- <u>Rejected Sample outcome</u>. This outcome occurs if
    - the observed value of <u>each</u> supported failure probability performance parameter from the transfer availability trial is equal to or better than its threshold, and
    - the observed value of the User Information Bit Transfer Rate from the transfer availability trial is worse than <u>the</u> User Information Bit Transfer Rate threshold and the performance is attributed to the user.[25]

---

[24]Since the supported performance parameters are bit-oriented, the performance parameters of the transfer function are not affected by systems that use different block lengths.

[25]See Section 4.2 of ANS X3.102-1992 for the procedure to distinguish between user and system nonperformance when the observed value of User Information Bit Transfer Rate is worse than the User Information Bit Transfer Rate threshold.

- Transfer Denial outcome. This outcome occurs if the observed performance of the transfer availability trial is worse than the threshold of any supported performance parameter, and the performance is attributed to the system.

Table 7 shows the three transfer availability trial outcomes for all combinations of observed values of the supported performance parameters in the transfer availability trial relative to their thresholds. In the table, "b" denotes observed values of supported performance parameters that are better than threshold, and "w" and "W" denote observed values of supported performance parameters that are worse than threshold when attributed to the system and the user, respectively. Since the Successful Transfer outcome and the Rejected Sample outcome can occur from only one combination of observed values relative to the threshold, the Transfer Denial outcome can occur from any of the remaining 14 of the $2^4 = 16$ combinations of observed values relative to the threshold.

2. Define Transfer Sample Transfer Performance Parameters

Three performance parameters are associated with the transfer sample transfer function: User Information Bit Transfer Rate, User Fraction of Input/Output Time, and Transfer Denial Probability.

a. User Information Bit Transfer Rate

This performance parameter is often called throughput. Since throughput should be measured only when the system is available, the following two definitions of User Information Bit Transfer Rate involve the quality of the transferred user information. Depending upon the desired length of the measurement, the standard defines this performance parameter as the rate of either a single transfer sample (that includes only those user information bits whose quality is perfect) or a series of transfer samples (that includes only those user information bits whose quality is perfect and that occur in transfer samples that are transfer availability trials and whose quality is equal to or better than the specified values of the supported performance parameters).

Table 7. Transfer Availability Trial Outcomes and Their Defining Conditions

	TRANSFER AVAILABILITY TRIAL OUTCOMES																	
SUPPORTED PARAMETERS	Successful Transfer	Rejected Sample	Transfer Denial															
Bit Error Probability	b	b	b	b	b	b	b	b	W	W	W	W	W	W	W	W
Bit Loss Probability	b	b	b	b	W	W	W	W	b	b	b	b	W	W	W	W
Extra Bit Probability	b	b	W	W	b	b	W	W	b	b	W	W	b	b	W	W
UIBTR	b	W	b	W	b	W	b	W	b	W	b	W	b	W	b	W

(i.) Single Transfer Sample

The User Information Bit Transfer Rate can be measured from a single transfer sample; its Input/Output Time may be short, providing a "short-term" measure of throughput, but it need not be. Since throughput should be measured only when the system is available, only Successful Bit Transfer attempts in the transfer sample are included.

User Information Bit Transfer Rate = (number of Successful Bit Transfer outcomes) / (Input/Output Time)

This version of the User Information Bit Transfer Rate uses the Successful Bit Transfer outcome, which is defined by the bit transfer function (and does not depend upon the supported performance parameters). The Input/Output Time is associated with the single transfer sample, which is defined by the transfer sample transfer function.

(ii.) Series of Transfer Samples

To obtain a "long-term" measure of throughput, the User Information Bit Transfer Rate should be measured over a number of transfer samples - the more the better. Since throughput should be measured only when the system is available, this version counts only Successful Bit Transfer outcomes in those transfer samples that are transfer availability trials and also have the Successful Transfer outcome.[26]

User Information Bit Transfer Rate = (number of Successful Bit Transfer outcomes in all Successful Transfer outcomes) / (Input/Output Time)

This version of the User Information Bit Transfer Rate uses the Successful Bit Transfer outcome which is defined by the bit transfer function and the Successful Transfer outcome which is defined by the transfer sample transfer function. The Input/Output Time is associated with the smallest transfer sample that includes all transfer samples used in the measurement; it is defined by the transfer sample transfer function.

[26]This "long-term" version of UIBTR is defined in terms of supported parameters, one of which is the UIBTR. However, this definition is not "circular" since the UIBTR supported parameter is the "short-term" version of UIBTR.

b. User Fraction of Input/Output Time

The User Fraction of Input/Output Time is the ancillary performance parameter for the User Information Bit Transfer Rate. Either the Input Time or the Output Time may be influenced by the user. For any transfer sample, the user-responsible Input/Output Time and the Input/Output Time are defined as follows:

- User-Responsible Input/Output Time is the larger of the User-Responsible Input Time or the User-Responsible Output Time:
  - User-Responsible Input Time is the portion of Input Time for which the user is responsible (e.g., a keyboard operator typing slower than the maximum rate at which the system can accept characters would be responsible for the input delay), and
  - User-Responsible Output Time is the portion of Output Time for which the user is responsible (e.g., an application program delaying acceptance of user information blocks until previously delivered blocks have been processed would be responsible for the output delay).
- Input/Output Time is defined from the transfer sample transfer function.

Similar to its primary performance parameter, the User Fraction of Input/Output Time can be measured as a ratio of values associated with either a single transfer sample or a series of transfer samples.

(i.) Single Transfer Sample

The User Fraction of Input/Output Time can be measured from a single transfer sample; its Input/Output Time may be short, providing a "short-term" measure, but it need not be.

User Fraction of Input/Output Time = (User-Responsible Input/Output Time) / (Input/Output Time)

The User-Responsible Input/Output Time and the Input/Output Time are both associated with the single transfer sample.

(ii.) Series of Transfer Samples

To obtain a "long-term" measure, the User Fraction of Input/Output Time should be measured over a number of transfer samples - the more the better. Since it should be measured only when the system is available, the associated transfer samples must be those that are transfer availability trials and also have the Successful Transfer outcome.

User Fraction of Input/Output Time = (User-Responsible Input/Output Time) / (Input/Output Time)

The Input/Output Time is associated with the smallest transfer sample that includes all transfer samples used in the User Information Bit Transfer measurement.

c. Transfer Denial Probability

The Transfer Denial Probability is measured from a series of transfer availability trials. The Transfer Denial outcome is determined by the supported performance parameters (i.e., the three bit-oriented probability performance parameters and the "short-term" version of the User Information Bit Transfer Rate).

Transfer Denial Probability = (number of Transfer Denial outcomes) / (number of transfer availability trials)

1.3.3 Disengagement Function

A separate disengagement function is associated with each end user. Since the processes are identical for each end user, it is not necessary to distinguish them here.

A. Determine Disengagement Outcomes

Figure 7 illustrates the time criteria for disengagement outcomes. There are three possible outcomes for each disengagement function:

- Successful Disengagement outcome. This outcome occurs if the Disengagement Confirmation reference event occurs no later than the Disengagement Timeout.

- Disengagement Denial outcome. This outcome occurs if the Disengagement Confirmation reference event occurs later than the Disengagement Timeout, and the excessive delay is attributed to the system.

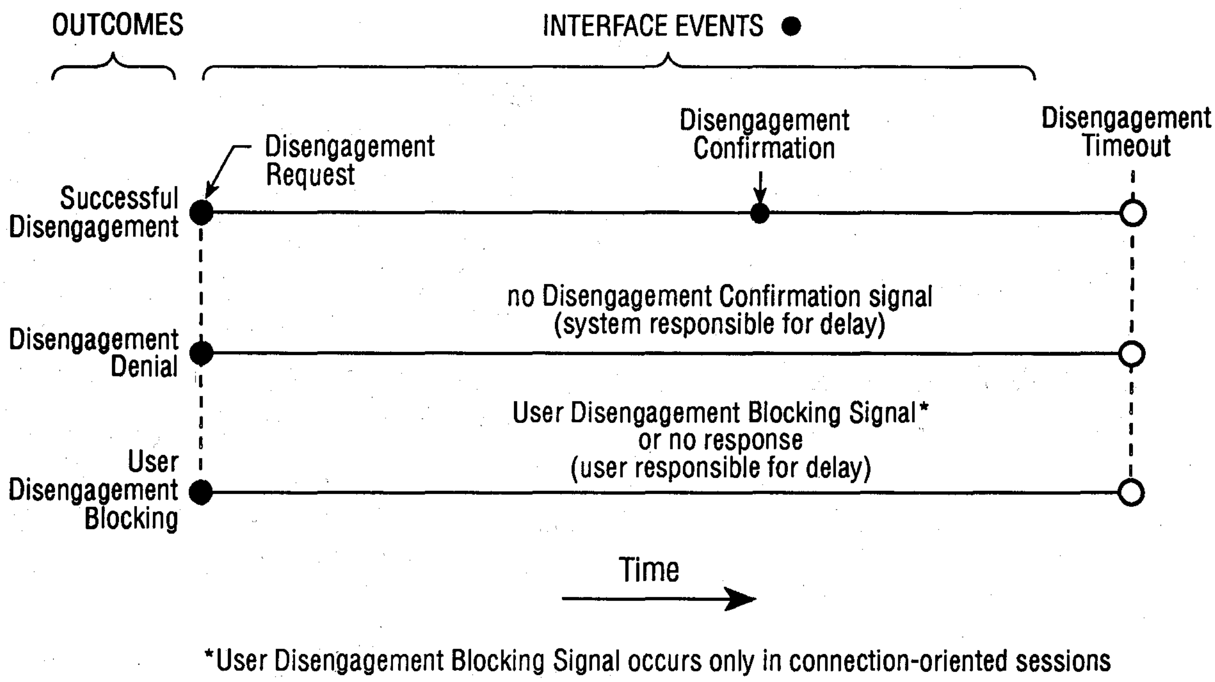


Figure 7. Scheme that determines disengagement outcomes according to ANS X3.102.

- <u>User Disengagement Blocking outcome</u>. This outcome occurs if either
  - a User Disengagement Blocking Signal reference event is issued in a connection-oriented session no later than the Disengagement Timeout (e.g., a "close request" signal that is ignored by the other user), or
  - Disengagement Timeout occurs, and the excessive delay is attributed to the user.

B. Define Disengagement Performance Parameters

Time performance parameters can be calculated from a single disengagement attempt whose outcome is Successful Disengagement (although it is usual to calculate them as an average of several Successful Disengagement outcomes). Failure probability performance parameters are defined as ratios of the number of failure outcomes to the number of attempts. For the disengagement function, the standard defines failure probability performance parameters as the ratio of the number of failure outcomes to the "total disengagement attempts included in a disengagement parameter determination." This total is the sum of all disengagement attempts except those whose outcome is User Disengagement Blocking; for convenience, the total is referred to here as the "number of disengagement attempts."

Three performance parameters are associated with the disengagement function: one primary time performance parameter, its ancillary performance parameter, and one failure probability performance parameter.

1. Disengagement Time

This performance parameter is calculated from Successful Disengagement outcomes only.

Disengagement Time = average [(time of Disengagement Confirmation) - (time of Disengagement Request)]

where

- Disengagement Confirmation is a primary reference event, and
- Disengagement Request is a primary reference event.

2. User Fraction of Disengagement Time

This performance parameter is calculated from Successful Disengagement outcomes only.

User Fraction of Disengagement Time = (average User-Responsible Disengagement Time) / (Disengagement Time)

where

- User-Responsible Disengagement Time is that portion of Disengagement Time for which the user is responsible,[27] and
- Disengagement Time has just been defined.

3. Disengagement Denial Probability

Disengagement Denial Probability = (number of Disengagement Denial outcomes) / (number of disengagement attempts)

The "number of disengagement attempts" paraphrases the "total disengagement attempts included in a disengagement parameter determination" as stated in the standard.

[27]User-Responsible Disengagement Time will depend upon the user and the system. For example, in a system employing a "four-way handshake", the user must respond to the system-generated Disengagement Request reference event before the system can disengage.

2. ANS X3.141 (MEASUREMENT METHODS)

2.1 Introduction

This standard specifies uniform methods to measure the ANS X3.102 performance parameters. The uniform methods may be used to obtain performance parameter estimates at any pair of digital interfaces connecting a data communication system to its end users. This standard specifies four phases: experiment design, data extraction, data reduction, and data analysis. Figure 8 is a structured design diagram showing the relationship among the four phases. The dashed line indicates that examination of the analyzed data might prompt redesign of the experiment.

2.2 Experiment Design

Even though experiments are somewhat exploratory, their results can be maximized if all applications are anticipated.

Experiment design is used to obtain measurements that provide

- accuracy (i.e., unbiased estimates with stated precision),
- clearly defined applicability, and
- efficient use of resources.[28]

---

[28]The following concepts from probability and statistics are discussed in this section:

- A <u>trial</u> is a single attempt to perform a data communication function.
- A <u>population</u> is the set of all possible trials of interest in the experiment.
- A <u>sample</u> is a subset of trials from the population.
- A <u>factor</u> is a condition existing during the experiment that may affect one or more performance parameters (e.g., day of the week).

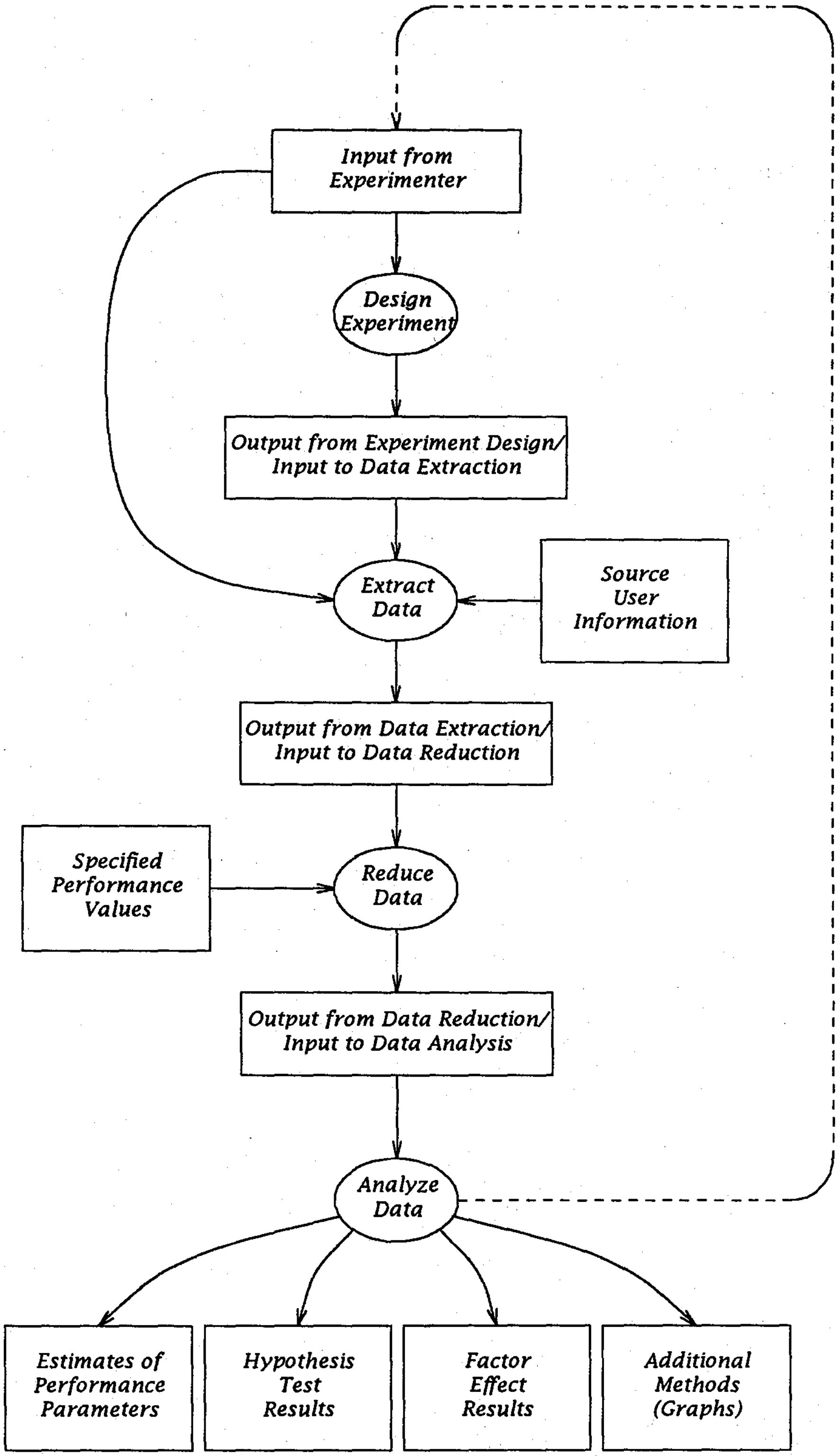


Figure 8. Structured design diagram of ANS X3.141 specifications.

As applied to data communication services, experiment590Xdesignnsists of seven steps.

2.2.1 Determine the Objectives of the Experiment

The first step in experiment design is to determine how the measured data will be used. Experiment objectives are defined somewhat broadly. A few examples are acceptance, characterization, and optimization. Within the objectives lie some specific technical values to be achieved and some business consequences such as pricing.

To achieve these objectives, the standard requires one or more of three analyses: estimates of performance parameters, tests of hypotheses, and analyses of the effects of factors. These analyses are discussed in Section 2.5.[27]

2.2.2 Select the Performance Parameters to be Measured

Any subset of the ANS X3.102 performance parameters can be measured. Ideally, all performance parameters required by the experiment objectives are determined. However, cost, time, and the availability of facilities might limit the number of performance parameters that can be measured. Sometimes it is convenient to measure a subset of the access and disengagement performance

---

28(cont'd)

- A <u>level</u> is a state of a factor. It might be qualitative (e.g., Thursday is a qualitative level of the factor, day of the week) or quantitative (e.g., 512 bytes is a quantitative level of the factor, block size).

- A <u>combination of levels</u> (i.e., one level from each factor) is the condition under which a sample is obtained to estimate the performance parameters.

- A <u>confidence level</u> is a value, usually expressed as a percent, that describes the likelihood that an interval, calculated from the sample, will contain the true value of the estimated performance parameter (e.g., 95% is a confidence level).

- A <u>significance level</u> is a specified value, usually expressed as a percent, for hypothesis testing that can be expressed as the complement of the confidence level (e.g., 5% is a significance level).

parameters at the same time and/or a subset of the user information transfer performance parameters at the same time. If possible, performance parameters from the same communication function should be measured under the same or similar conditions.

2.2.3 Define the Population

To define the population (i.e., all attempts to perform a data communication function), it is necessary to also define

- the period(s) over which the experiment is to be conducted,
- the set of all end user-pairs that is of interest to the experiment objective,
- the characteristics of all end user-pairs to which service is provided,
- the characteristics of the user-system interfaces,
- the session profile (See Figure 23),
- the reference events,
- the type of data communication session, and
- specified performance values for certain performance parameters (e.g., values that specify the performance time after which a primary time performance parameter is considered to be a failure).

2.2.4 Determine the Combinations of Levels to be Tested

Generally, the number of factors and levels that could influence data communication services and its users are too vast to be compiled. However, it is necessary to list the factors and levels that are relevant to the experiment objectives and, further, to determine the combinations of levels that are expected to influence performance. For efficiency, the following three principles should be followed in selecting the combinations of levels:

- <u>Select the Minimum Number of Levels</u>. Levels of factors should be selected only if their effects on the experiment are required by the experiment objectives.

- Test all Combinations of Levels and Replicate. Trials for each combination of levels should be obtained, and they should be replicated, if possible, to reveal unknown factors.

- Select Levels to Maximize Accuracy. If all combinations of levels cannot be tested, select those that will achieve the maximum accuracy from those causing the most variation.

Various designs, such as full factorial and fractional factorial designs, exist from which interactions among factors can be determined.

2.2.5 Determine the Population Sample

From the population of performance trials, select a representative sample and determine the sample size.

A. Select a Representative Sample

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. Randomization should be obtained if possible. However, practical constraints, such as the following, sometimes preclude (complete) randomization:

- Efficient Extraction. For efficiency, data extraction procedures may require trials to be obtained in a certain, nonrandom way.

- Service Interruption. Sometimes data communication service can not be interrupted during the experiment.

- Inhomogeneous Samples. Sometimes certain combinations of levels have values that are similar to each other but different than those of other combinations. The combinations with similar values can be grouped (a procedure called blocking), and the sample size from each combination of the block should be similar.

- Unbalanced Testing. Sometimes a random sample would be unacceptable because some combinations of levels would be tested too many times or not enough (resulting in unbalanced testing).[29]

---

[29]That is, randomization is restricted when using certain blocking designs, such as randomized blocks, balanced incomplete blocks, and Latin squares.

- Dependence. Dependence may exist among trials, particularly among those observed closely in time or location. Dependence usually increases the number of trials required to achieve a specified precision; dependence should be considered by methods such as described here and in Miles (1984).

Unavoidable departures from randomization may be rectified in one of two ways:

- Three-Stage Sampling. Prior to sampling, impose constraints on the random selection of performance trials. For example, obtain samples according to
    - geographical area,
    - end users within each geographical area, and
    - performance trials for each end user (within each geographical area).
- Balanced Sampling. Obtain samples without restriction, but reject statistically unbalanced samples.

B. Determine the Sample Size

The sample size for the most important performance parameter may be dictated by

- a specified precision (for a specified confidence level or significance level), or
- a specified time and/or budget (in which case an unspecified precision will be achieved for a specified confidence level or significance level).

The confidence level or significance level is determined from the technical or business consequences (due to the experiment objective).

2.2.6 Select the Levels

The number of combinations of levels may be quite large. For example, if four factors are thought to affect the experiment, and they have 3, 2, 4, and 3 levels, the number of combinations of levels is

$$3 \times 2 \times 4 \times 3 = 72.$$

If the number of combinations of levels is too large, a random subset must be selected. (If all combinations of levels are measured, the design is called a factorial design, otherwise it could be one of many fractional factorial designs.)

The steps in Sections 2.2.2 through 2.2.5 are intended to satisfy a range of applications, whereas this step is intended to achieve a favorable combination of efficiency and accuracy.

2.2.7 Determine the Mathematical Model

It is often useful, but not essential, to describe the experiment design by a mathematical model. For example, the simple mathematical model,

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

relates Y_i, the measured value of the ith observation, μ, the population mean of the performance parameter, and ϵ_i, the experimental error in the ith observation.

2.3 Data Extraction

This phase specifies how performance data are to be extracted from the interfaces. An independent interface monitor for each measured interface extracts performance data by

- collecting interface events,
- processing interface events, and
- recording reference events (i.e., those system-independent interface events having performance significance).

These three steps are discussed in the following three sections.

2.3.1 Collect Interface Events

During a performance period, the interface monitor detects all signals crossing the interface, interprets them as a sequence of interface events, and determines their times of occurrence. There are two types of interfaces: user-system interfaces and subsystem-subsystem interfaces:

PART I. ANS X3.141 SPECIFICATIONS

A. User-System Interfaces

This type of interface is defined in Section 1.2.3.

B. Subsystem-Subsystem Interfaces

This standard also specifies how to measure the performance of system elements that terminate at those digital interfaces that comprise a portion of the data communication system. For example, three types of subsystem-subsystem interfaces are

- the physical interface between DTE and DCE,[30]
- the functional interface between adjacent layers of protocols (e.g., between the network layer and the transport layer in the Open Systems Interconnection (OSI) Reference Model), and
- the functional interface between adjacent networks. (It exist at a gateway node between the internetwork protocols at the network layer in the OSI Reference Model.)

2.3.2 Process Interface Events

Interface events are processed by identifying them and categorizing them according to their function. Primary reference events are interface events that define primary data communication functions. An ancillary reference event is an interface event that defines the entity responsible for generating the next reference event and identifies the entity responsible for nonperformance. An interface event can be a primary reference event, an ancillary reference event, both, or neither. Figure 9 is a Venn diagram depicting the relationship among types of interface events: The set of primary reference events and the set of ancillary reference events are subsets of the set of interface events, and these two sets of reference events have an intersection of reference events that are both primary and ancillary. An example of an interface event that is a primary reference event only is the End of Block Transfer reference event that occurs after the delivery of user information in the absence of flow control (responsibility at the destination interface is not changed). An example of an

---

[30]Data terminal equipment and data communication equipment have the acronyms DTE and DCE, respectively.

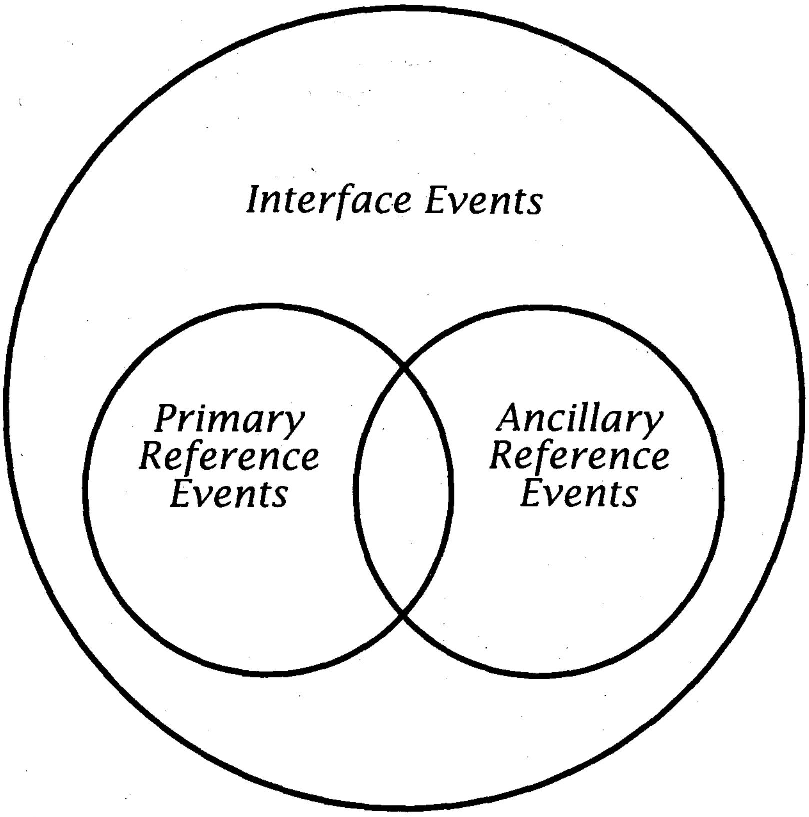


Figure 9. Venn diagram of interface events, primary reference events, and ancillary reference events.

event that is an ancillary event only is the issuance of a dial tone by the public switched telephone network. An example of an interface event that is both a primary reference event and an ancillary reference event is a calling user's issuance of an X.25 **Call Request** packet: It is an Access Request primary reference event, and it also relieves both the system and the user of responsibility for producing the next event at the local interface. An interface event that is neither a primary reference event nor an ancillary reference event is any other communication between the system and the end user.

A. Primary Reference Events

Primary reference events define primary data communication functions. The ten primary reference events are listed and defined in Table 3.

B. Ancillary Reference Events

Ancillary reference events define the entity responsible for generating the next interface event at a specific interface, and they identify the entity responsible for nonperformance in an attempt to perform a data communication function. Hence, they help define ancillary performance parameters (i.e., those performance parameters that measure the fraction of the total performance delay that is attributable to the user). Ancillary performance parameters are listed in italics in Table 4.

After an interface event occurs at an interface (arbitrarily call it the local interface), one or more entities may be responsible for producing the next interface event at either the local or the remote interface. Theoretically, the four entity-interface responsibilities are

- local user responsibility at the local interface,[31]
- system responsibility at the local interface,[32]
- remote user responsibility at the remote interface, and

[31]For example, responsibility passes to the local user at the local interface after the system issues a dial tone.

[32]For example, responsibility passes to the system at the local interface after the calling user causes an off-hook action.

- system responsibility at the remote interface.[33]

However, the third entity-interface responsibility cannot exist: The remote user cannot be responsible for producing the <u>next</u> reference event because a reference event must occur at the remote interface before the remote user can be responsible for producing a reference event.

The three remaining entity-interface responsibilities and two levels of responsibility (i.e., yes and no) produce the $2^3 = 8$ combinations that are listed in Table 8.[34] (The seventh and eighth combinations cannot exist - as indicated by the absence of shading - because the user and the system cannot <u>both</u> be responsible for the next reference event at an interface.)

Table 8. Combinations of Responsibilities for the Three Entity-Interface Pairs

	LOCAL USER RESPONSIBLE AT LOCAL INTERFACE?	SYSTEM RESPONSIBLE AT LOCAL INTERFACE?	SYSTEM RESPONSIBLE AT REMOTE INTERFACE?
1	No	No	No
2	No	No	Yes
3	No	Yes	No
4	Yes	No	No
5	No	Yes	Yes
6	Yes	No	Yes
7	Yes	Yes	No
8	Yes	Yes	Yes

[33]For example, in the X.25 protocol, a **Call Request** packet temporarily relieves both the calling user and the system of responsibility for generating the next interface event at the calling (local) interface because the next interface event (delivery of the **Incoming Call** packet) occurs at the remote interface.

[34]If neither entity at an interface is responsible for producing the next event, the responsibility could be considered undefined.

To generate the sequence of interface events during a data communication session (as depicted in a session profile), it is necessary to identify - following each reference event - the entity responsibile for producing the next reference event at each interface.

2.3.3 Record Reference Events

To measure performance parameters, the interface monitor should record

- the reference events,
- the times of their occurrence, and
- sufficient information about the transferred and received user information to determine transfer failures. The procedure depends upon whether the user information is binary or not:
  - All Binary. If user information is transmitted and received as binary user information, the monitor shall record both sets and map the source user information into the destination user information (when the two differ).
  - Not All Nonbinary. If both/either sets of user information are transmitted and/or received as nonbinary user information, the monitor shall generate the binary representations, record both sets, and map the source user information into the destination user information (when the two differ).

2.4 Data Reduction

This phase implements the performance parameter definition process of ANS X3.102 (Section 1.3): It converts extracted data to performance trial values. In all cases, outcomes that are attributable to user delays are excluded from performance evaluation.

2.4.1 Calculate Access Performance Parameters

Figure 5 shows the access outcomes as they depend upon the relative order of interface events and the Access Timeout according to ANS X3.102. Figure 10 is a flowchart showing how to determine the outcome of an access attempt

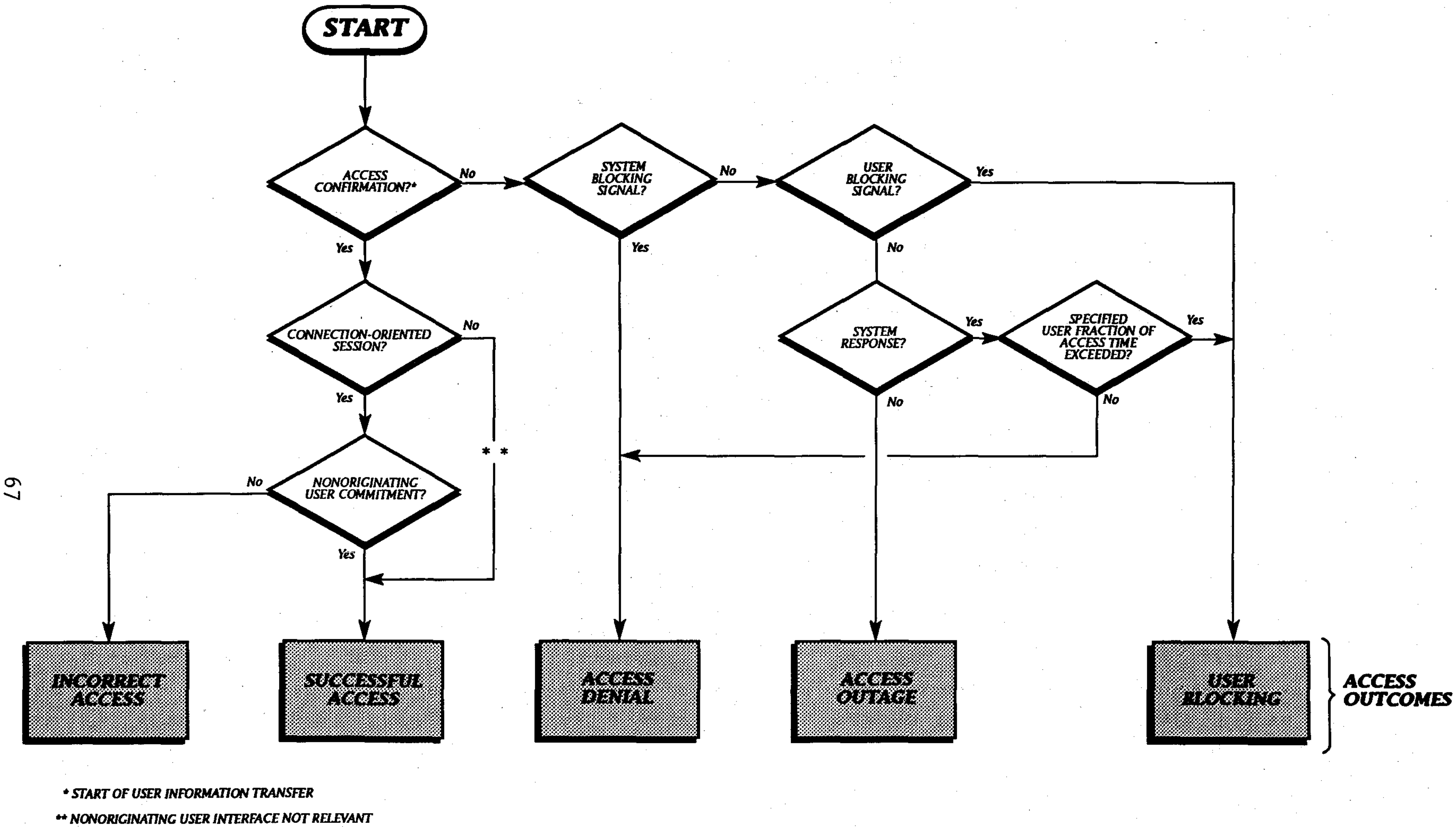


Figure 10. Flowchart that determines access outcomes according to ANS X3.141.

according to ANS X3.141. The outcomes are shown in shaded rectangles. The Access Confirmation reference event (or, equivalently, the Start of Block Transfer reference event) must occur for an access attempt to have a Successful Access outcome.[35] Also, if the specified User Fraction of Access Time is exceeded, the attempt is classified as the User Blocking outcome.

2.4.2 Calculate User Information Transfer Performance Parameters

The reference events and the transferred user information are examined to calculate the user information transfer performance parameters. This function can be conveniently described by dividing it into three functions: Bit transfer, block transfer, and transfer sample transfer (although this third function is not defined as a function by either the original ANS X3.102 or its revision).

A. Bit Transfer Outcomes

Table 5 lists bit transfer outcomes as defined by ANS X3.102. It shows that the outcomes are determined by the disposition of the bits and the block:

- Disposition of Bits. The bit transfer outcomes depend upon the accuracy and the end user location of the bits.
- Disposition of Block. The bit transfer outcomes depend upon the timeliness of the block transfer and the entity responsible for the delay when it exceeds Block Transfer Timeout.

ANS X3.141 formalizes the two conditions for the disposition of the bits and implements the two conditions for the disposition of the block to determine the bit transfer outcomes.

---

[35]ANS X3.141 states that the Start of Block Transfer reference event ends the access function, but the Access Confirmation reference event has subsequently been defined by the revised version of ANS X3.102 to end the access function. Either reference event can be considered as ending it.

1. Accuracy and End User Location of Bits.

ANS X3.141 formalizes the first pair of conditions by defining Bit Comparison Outcomes (BCOs).[36] Specifically, the source and destination blocks of user information are compared (a process called data correlation) to detect correct, incorrect, undelivered, and extra BCOs:

- Correct BCO. Corresponding bits exist in the source and the destination user information records, and their binary values are the same.
- Incorrect BCO. Corresponding bits exist in the source and the destination user information records, but their binary values differ.
- Undelivered BCO. A bit in the source user information record has no counterpart in the destination user information record. This outcome occurs if the End of Block Transfer reference event does not occur.
- Extra BCO. A bit in the destination user information record has no counterpart in the source user information record. This outcome occurs if the Start of Block Transfer reference event does not occur.

Figure 11 is a Venn diagram showing the four BCOs as they depend upon end user location and accuracy (when corresponding bits are present in both end user locations), and Figure 12 is a two-part schematic diagram illustrating BCOs within a sequence of transferred bits: Figure 12.a illustrates undelivered bits, and Figure 12.b illustrates correct, incorrect, and extra bits.

2. Timeliness and Entity Responsibility with Respect to Blocks

After the bits in each block have been correlated, ANS X3.141 implements the two conditions (i.e., timeliness and entity responsibility) for the disposition of the block to determine bit transfer outcomes. Figure 13 is a flowchart that illustrates this implementation. Those blocks whose Block Transfer Time exceeds the Block Transfer Timeout or contain all undelivered bits are examined to determine if the User Fraction of Block Transfer Time exceeds the

[36]Note that the BCOs are independent of the two conditions for the disposition of the block: time and entity responsibility.

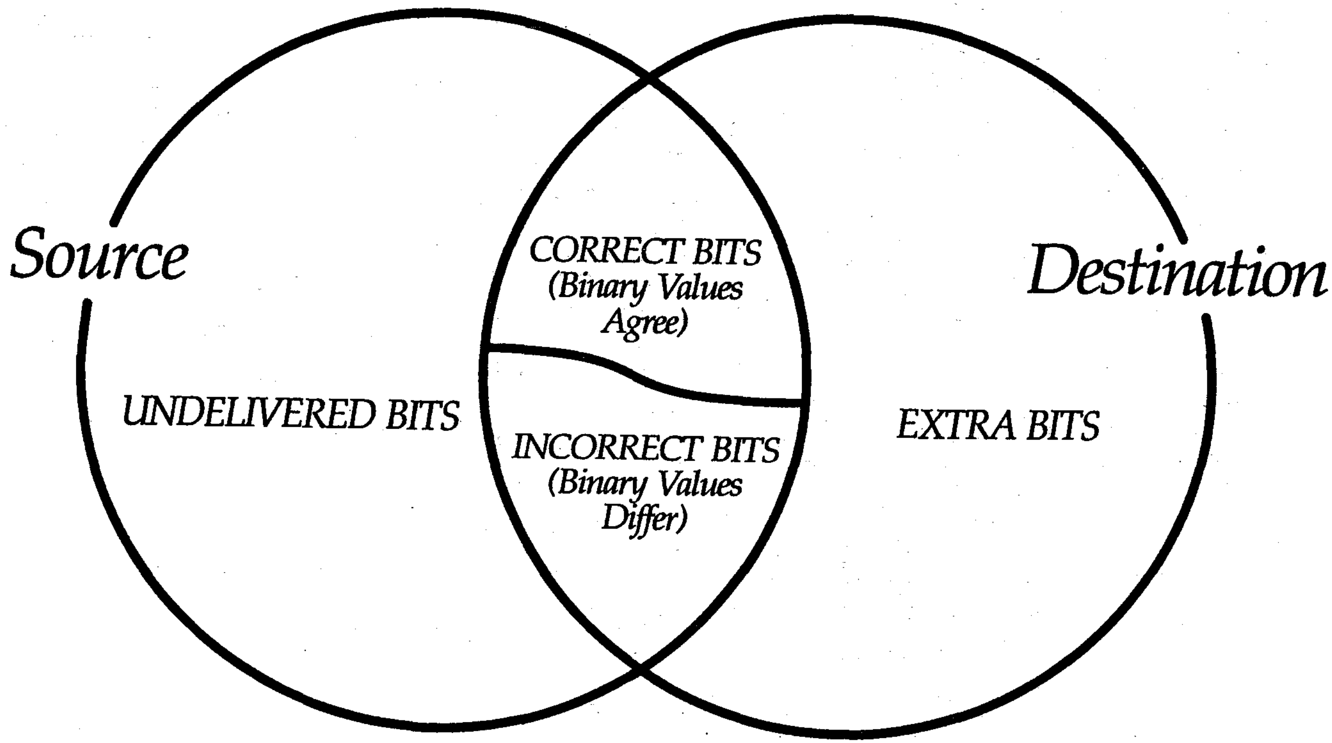


Figure 11. Venn diagram of Bit Comparison Outcomes (BCOs).

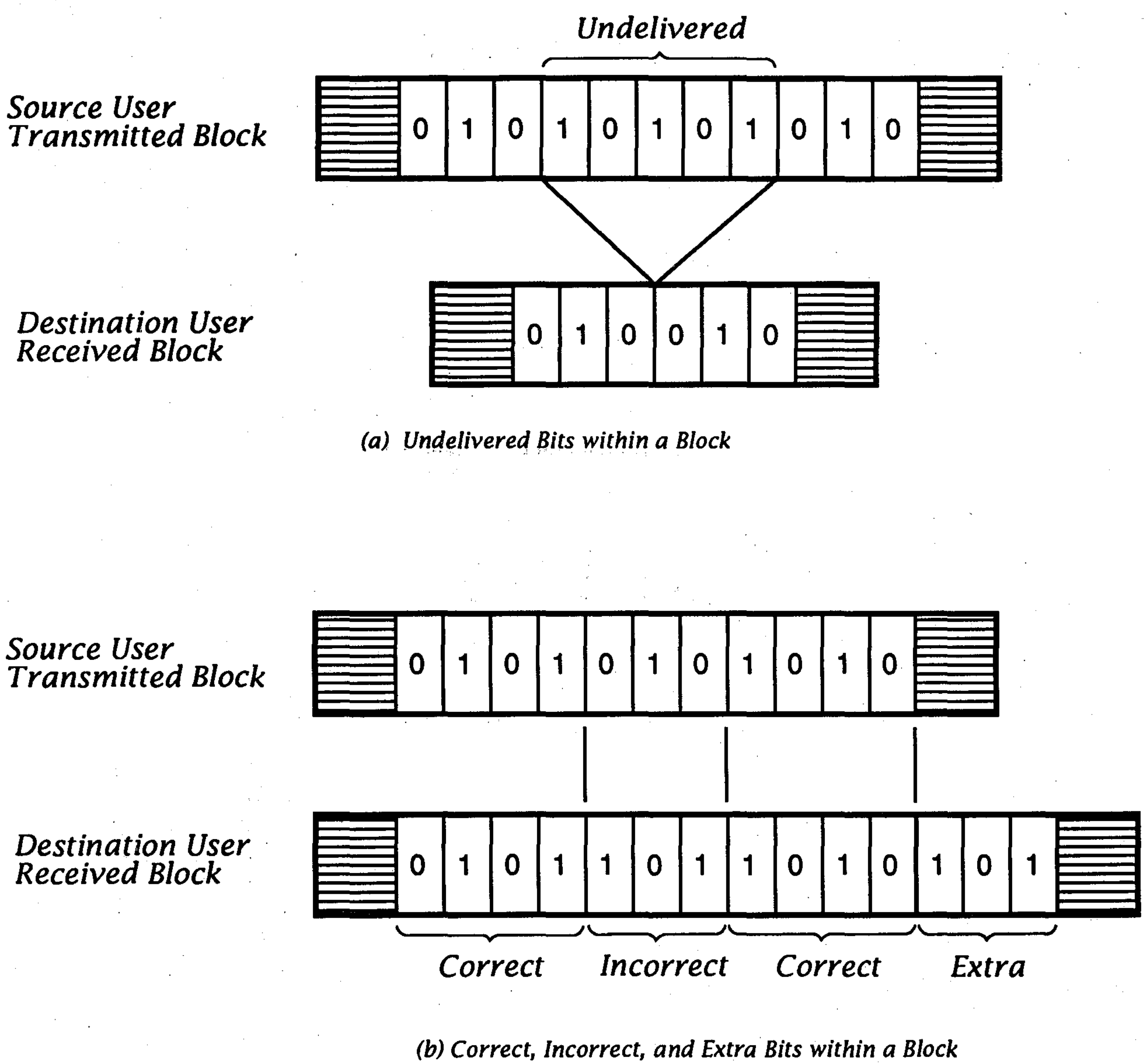


Figure 12. Schematic diagram of Bit Comparison Outcomes (BCOs).

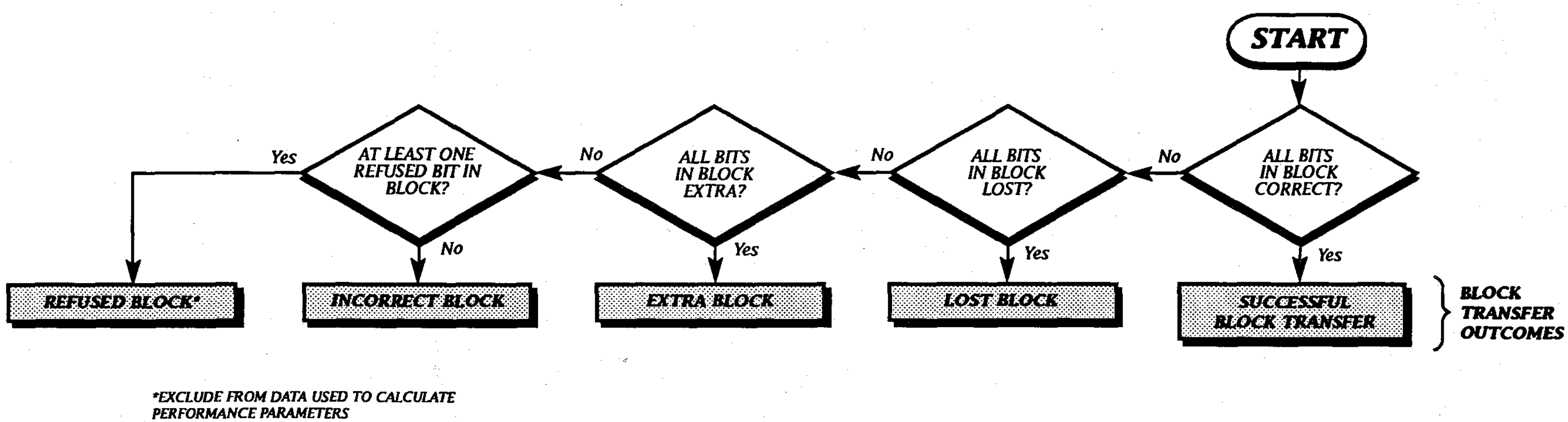


Figure 14. Flowchart that determines block transfer outcomes according to ANS X3.141.

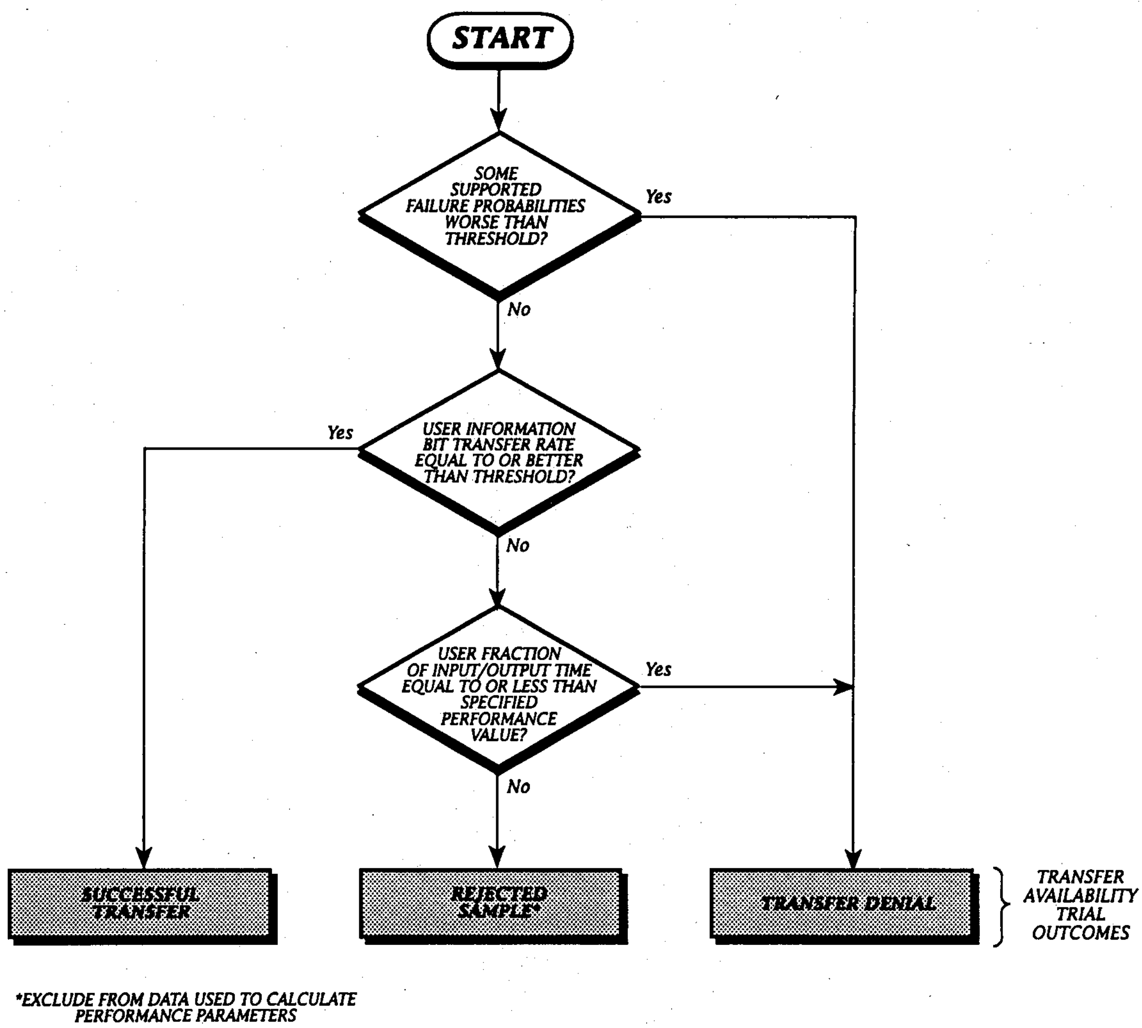


Figure 15. Flowchart that determines transfer availability trial outcomes according to ANS X3.141.

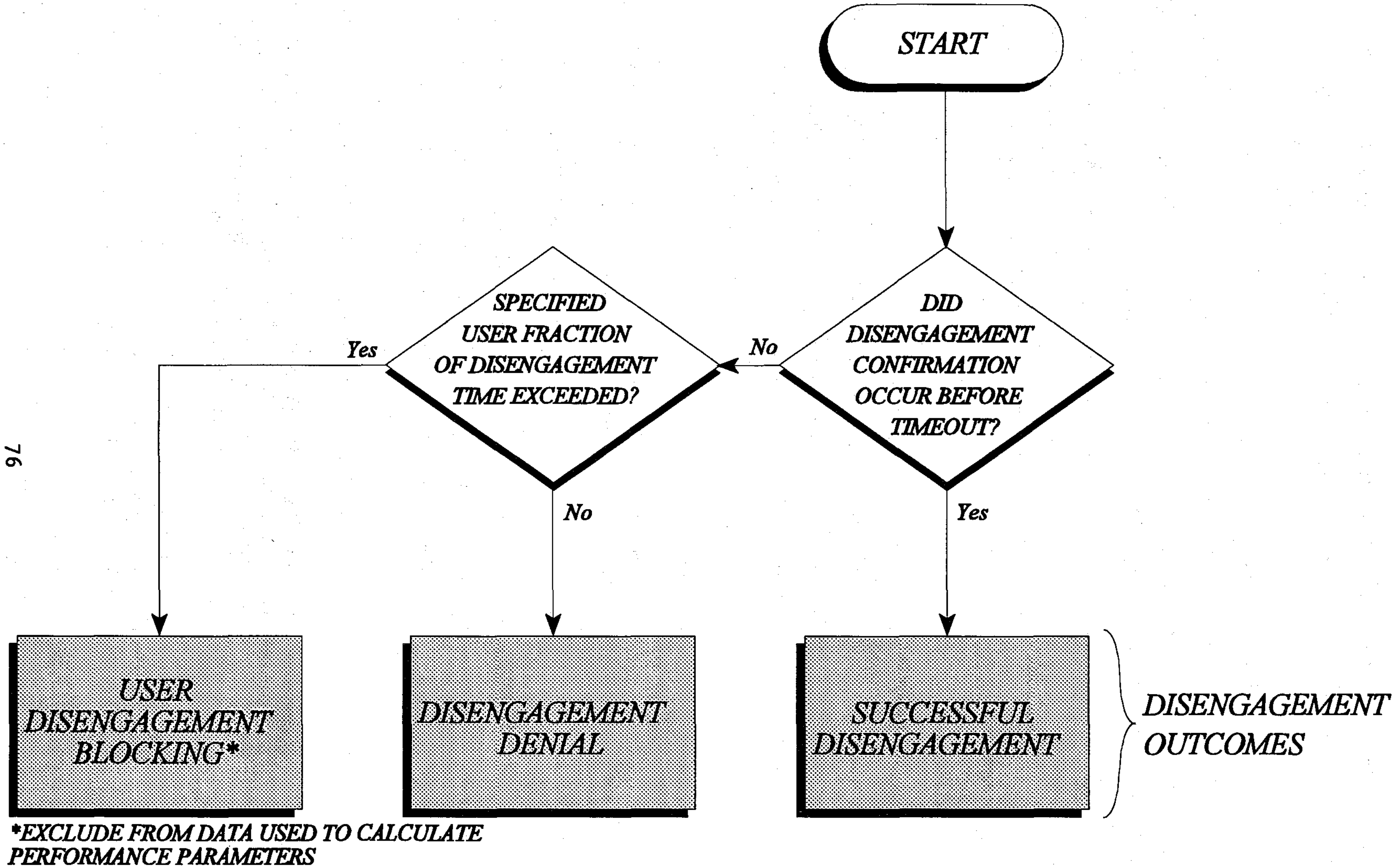


Figure 16. Flowchart that determines disengagement outcomes according to ANS X3.141.

Successful Disengagement and Disengagement Denial into separate outcomes for each end user. The possible outcomes are shown in the shaded rectangles.

2.4.4 Calculate User Performance Times

Entity performance times must be distinguished so they can be allocated between the user and the system. The ancillary performance parameters can be determined, and responsibility for timeouts can be assigned to either entity.

Figure 17 is a structured design diagram that shows the two processes that are necessary to estimate user performance times. One process consolidates the ancillary reference events from the two end users, and the other process allocates performance times.

A. Consolidate Ancillary Reference Events

An ancillary event history is obtained at each user-system interface. The history begins with the entity-responsibility state existing at the local interface prior to the first interface event, and it then includes, for each interface event,

- the ancillary event time,
- the entity-responsibility state for producing the next interface event at the local interface, and
- the responsibility effect at the remote interface.

The two ancillary event histories are consolidated to produce a single consolidated event history. Consolidation is necessary because the data from a single interface do not provide enough information to determine the user performance time.

Responsibility is based on two rules, depending upon the interface:

- <u>Responsibility at Local Interface</u>. An ancillary event at the local interface <u>determines</u> the subsequent responsibility at the local interface: local user responsibility, system responsibility, or responsibility undefined.
- <u>Responsibility at Remote Interface</u>. An ancillary event at the local interface <u>changes</u> the subsequent responsibility at the remote interface to system responsibility only if both

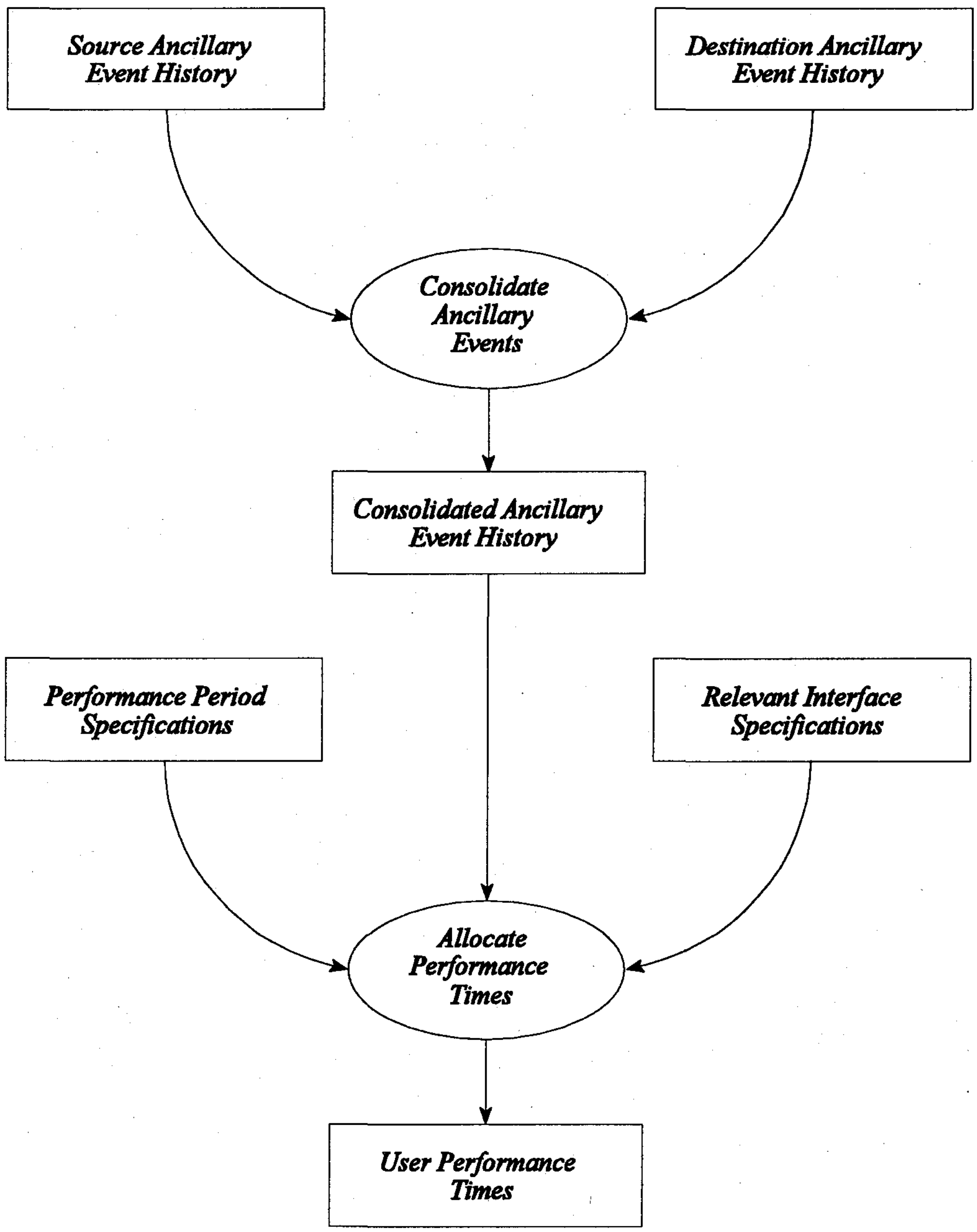


Figure 17. Structured design diagram of procedures to determine user performance times.

entities at the remote interface are waiting for that event (i.e., if responsibility is undefined).

B. Allocate Performance Times

The consolidated ancillary event history is examined to identify periods of user responsibility. Each primary data communication function begins with a primary reference event and ends with either a primary reference event or a timeout. The user performance time in a performance period is the sum of the user performance times during that period.

User performance times may be calculated during the following performance periods:[37]

- Access. User performance times may be calculated between the beginning and the end of an access attempt,
- Block Transfer. User performance times may be calculated between the beginning and the end of a block transfer attempt,
- Transfer Sample Transfer. User performance times may be calculated between the beginning and the end of the Input/Output Time of a transfer sample, and
- Disengagement. User performance times may be calculated between the beginning and the end of a source disengagement attempt or a destination disengagement attempt.

Table 9 lists the relevant user-system interfaces for performance periods under various performance conditions. Connection-oriented access requires a connection to be established before user information is transferred, and it requires a disconnection after transfer. Connectionless access allows user information to be transferred prior to the establishment of a connection. An independent disengagement attempt is one whose successful completion does not require a concurring response. A negotiated disengagement is one whose successful completion requires a concurring response from the user that does not cause the Disengagement Request reference event.

[37]No performance times are associated with the bit transfer function.

Table 9. Relevant Interfaces for Performance Time Allocation

PERFORMANCE TRIAL	PERFORMANCE CONDITION	RELEVANT INTERFACES
Access Attempt	Connection-Oriented	Source User-System and Destination User-System
	Connectionless	Source User-System
Bit Transfer Attempt	N/A	N/A
Block Transfer Attempt	All	Destination User-System
Transfer Sample	Responsibility Defined at Local Interface	Local User-System
	Responsibility Undefined at Local Interface	Remote User-System
Disengagement Attempt	Independent	Requesting User-System
	Negotiated	Source User-System and Destination User-System

When one interface is relevant, responsibility for the performance period is assigned to the entity identified in the consolidated event history.

When both interfaces are relevant (as in sessions involving either connection-oriented access or negotiated disengagement), responsibility for the performance period is assigned as follows:

- If the same entity is responsible at both interfaces, that entity is responsible.
- If a different entity is responsible at each interface, responsibility is said to be split, but the delay is attributed to the entity responsible for the longer delay.
- If an entity is responsible at one interface but responsibility is undefined at the other interface, the entity whose responsibility is defined is responsible.

- (It seems impossible that the responsible entities could be undefined at both interfaces.)

Figure 18 is a flowchart showing the method to calculate user performance time contributions. The contributions are shown in shaded rectangles.

2.5 Data Analysis

This phase analyzes the reduced performance data (i.e., the performance trials). It calculates statistical information from the experiment that can be used to accomplish the stated objective and should be reported. The first three sections of this phase describe estimation, hypothesis testing, and testing to determine the effects of factors. The fourth section discusses additional analyses that may be useful (but are not required).

2.5.1 Estimate Performance Parameters

Estimation characterizes the performance parameters of the data communication service for a single combination of levels of factors. Since a performance parameter estimate cannot be expected to equal the (true) population value because of sampling error, the estimate must be accompanied by some specified precision.

The specified precision should be expressed in terms of a confidence interval and an associated confidence level. The confidence interval should be calculated by a method similar to that described in Miles (1984) (i.e., a method that considers dependence using a first-order Markov chain).

While estimation is extremely important, it, alone, cannot be used to make decisions.

2.5.2 Test Hypotheses

A statistical hypothesis is an assumption about the distribution of a population, expressed in terms of specified hypothetical values for one or more population parameters.

An hypothesis test experiment is an experiment in which the validity of a statistical hypothesis is tested and, then, either accepted or rejected.

The decision to accept or reject a statistical hypothesis is made with some uncertainty if the (performance parameter) estimate is based on a finite sample;

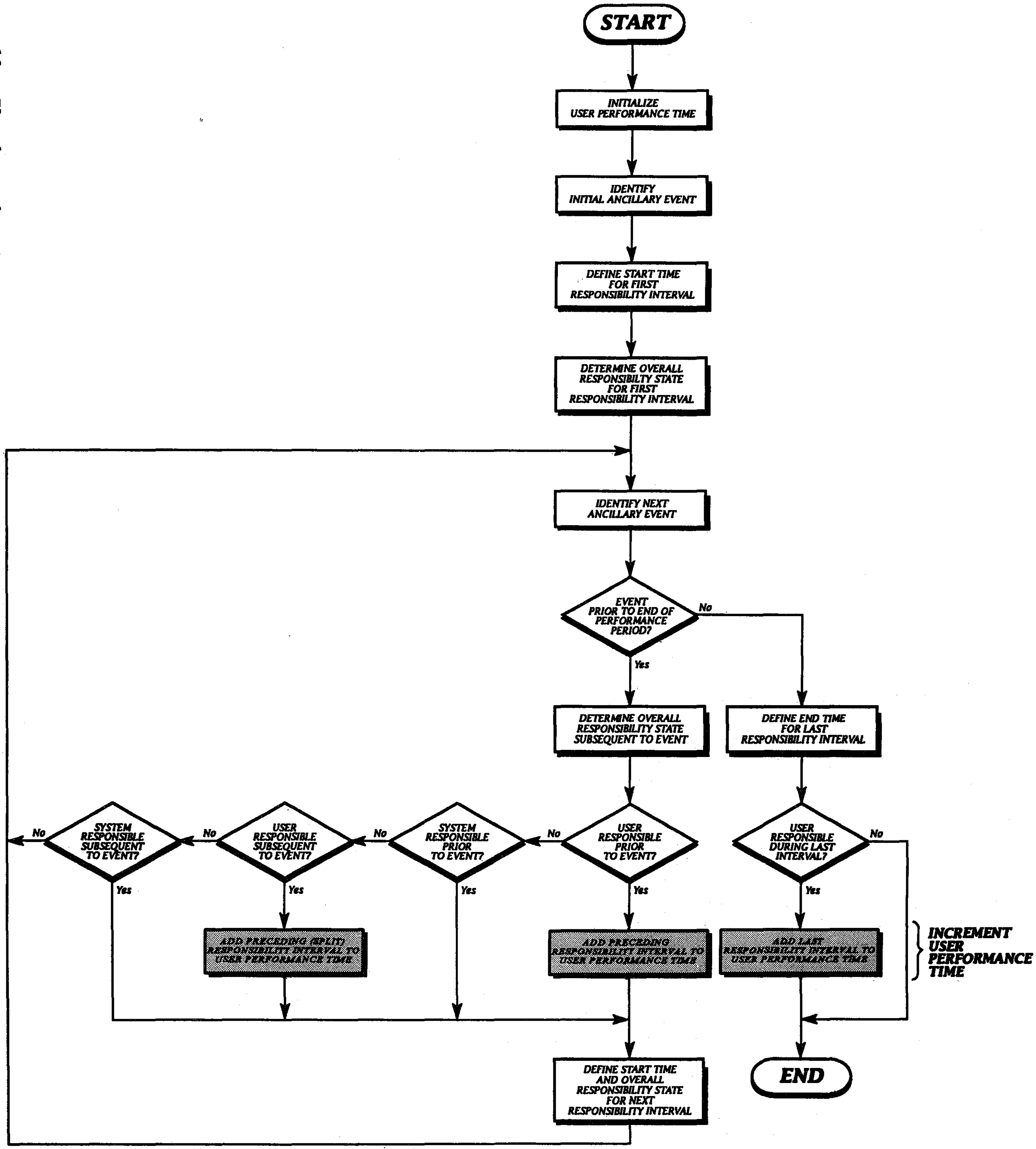


Figure 18. Flowchart that determines user performance time outcomes according to ANS X3.141.

the estimate can deviate from the population performance parameter value. The decision can result in two types of errors.

A. Type I Error

The simplest hypothesis test determines whether a significant difference exists between a performance parameter (estimated from trials that are obtained from a single combination of levels of variable conditions) and a specified hypothetical value.

The uncertainty of an hypothesis test experiment is expressed by its significance level α. This is the probability of rejecting the tested hypothesis when it is, in fact, true. This error is called a Type I error, and the probability, α, is the vendor's risk. The significance level should always be reported with the results of hypothesis test experiments.

1. Equality

The traditional statistical hypothesis is called the null hypothesis because its truth implies that no difference exists between the specified hypothetical and (true) population values.

The null hypothesis can be tested in the following way (as well as other ways):

- <u>Calculate</u>. Calculate a confidence interval from the trials. (If the null hypothesis is true, the calculated confidence interval includes the specified hypothetical value with probability 1 - α.)

- <u>Compare</u>. Compare the specified hypothetical value with the confidence interval. If the specified hypothetical value lies within the confidence interval, the null hypothesis can be accepted with a significance level α (i.e., probability of error). If the specified hypothetical value lies outside the confidence interval, the null hypothesis is rejected.

2. Inequality

An hypothesis that does not assume equality is called an alternative hypothesis.

The purpose of many hypothesis test experiments is to determine a positive hypothesis (i.e., that performance is equal to or better than - rather than equal to - a specified value). The hypothesis test can be applied in such experiments by simply halving the significance level. The resulting significance level expresses the probability that an observed value lies on the "performance side" of the confidence level.

The same approach can be used to test a negative hypothesis (i.e., that performance is worse than a specified value).

B. Type II Error

In some hypothesis test experiments, it may also be necessary or desirable to consider the probability of accepting a tested hypothesis when it is, in fact, false. This error is called a Type II error, and the probability, β, is the consumer's risk. The likelihood of such an error is determined by three variables: the significance level, α, of the hypothesis test experiment, the sample size, and the difference between the specified hypothetical and population values. Relationships among these variables may be determined from a curve of $1 - \beta$ (called a power curve).

2.5.3 Analyze Factor Effects

Data communication experiments are usually conducted under several combinations of levels. Performance data obtained from different combinations of levels can be compared to identify and quantify postulated effects of factors.

The analysis of variance (ANOVA) is a statistical technique that determines these effects. ANOVA can provide two analysis results:

- <u>Reveal No Effect</u>. It can reveal that a postulated factor has no significant effect on performance data (for the tested levels). Hence, the data can be combined - a result that simplifies analysis by eliminating unnecessarily distinguished levels and provides more precision by virtue of the larger, combined sample.

- <u>Reveal Some Effects</u>. It can reveal quantitative relationships among tested levels of a factor (thus, the data cannot be combined).

2.5.4 Use Additional Methods

Additional analysis methods that may be helpful are

- frequency distributions (e.g., histograms) and cumulative distributions (to determine statistical characteristics of the performance data),
- control charts (to determine if chronological performance data deviate from an allowable range of values), and
- regression analysis (to determine the effect of quantifiable levels of a factor on performance data).

PART II. NTIA DESIGN

Part II is an overview of the NTIA design of the two ANSI specifications. With the exception of the two optional performance parameters, Bit Misdelivery Probability and Block Misdelivery Probability, NTIA design implements ANS X3.141 and the original ANS X3.102. The design has been upgraded to reflect much of the revised ANS X3.102.

The NTIA design follows the four phases of ANS X3.141, and it includes an enhancement (henceforth called a phase) that implements some of the "additional methods" of analysis discussed in Section 2.5.4 of Part I. Specifically, this fifth phase graphs the primary time performance parameters.

Figure 19 is a structured design diagram that shows the NTIA design of the five phases. (It should be compared with Figure 8, a structured design diagram that shows the ANS X3.141 specifications.) Figure 19 includes the phases as well as input/output data and files that constitute

- input (i.e., files and data) to a phase from the experimenter,
- interphase input/output files,
- output files to the experimenter that are intended simply for information (e.g., the **nnnnovh.x**, **nnnnovh.r**, **nnnnhis.x**, and **nnnnhis.r** files contain reference event data and are output from the data extraction phase, and the **SUM** files contain assessment summaries and are output from the data reduction phase),[38] and
- output (i.e., files and data) to the experimenter from a phase that are intended to satisfy the experiment objective.

The diagram demonstrates the NTIA design for a single test (as defined by a single combination of levels of variable conditions) and multiple tests (as defined by multiple combinations of levels of variable conditions). Each phase and each interphase input/output file list the number of the section in <u>this volume</u> in which the phase and file are discussed. Single tests culminate with data analysis (described in Section 6.1.1) and data display (described in

---

[38]Many files are not shown in this diagram because they are of lesser importance, don't transcend phases, etc.

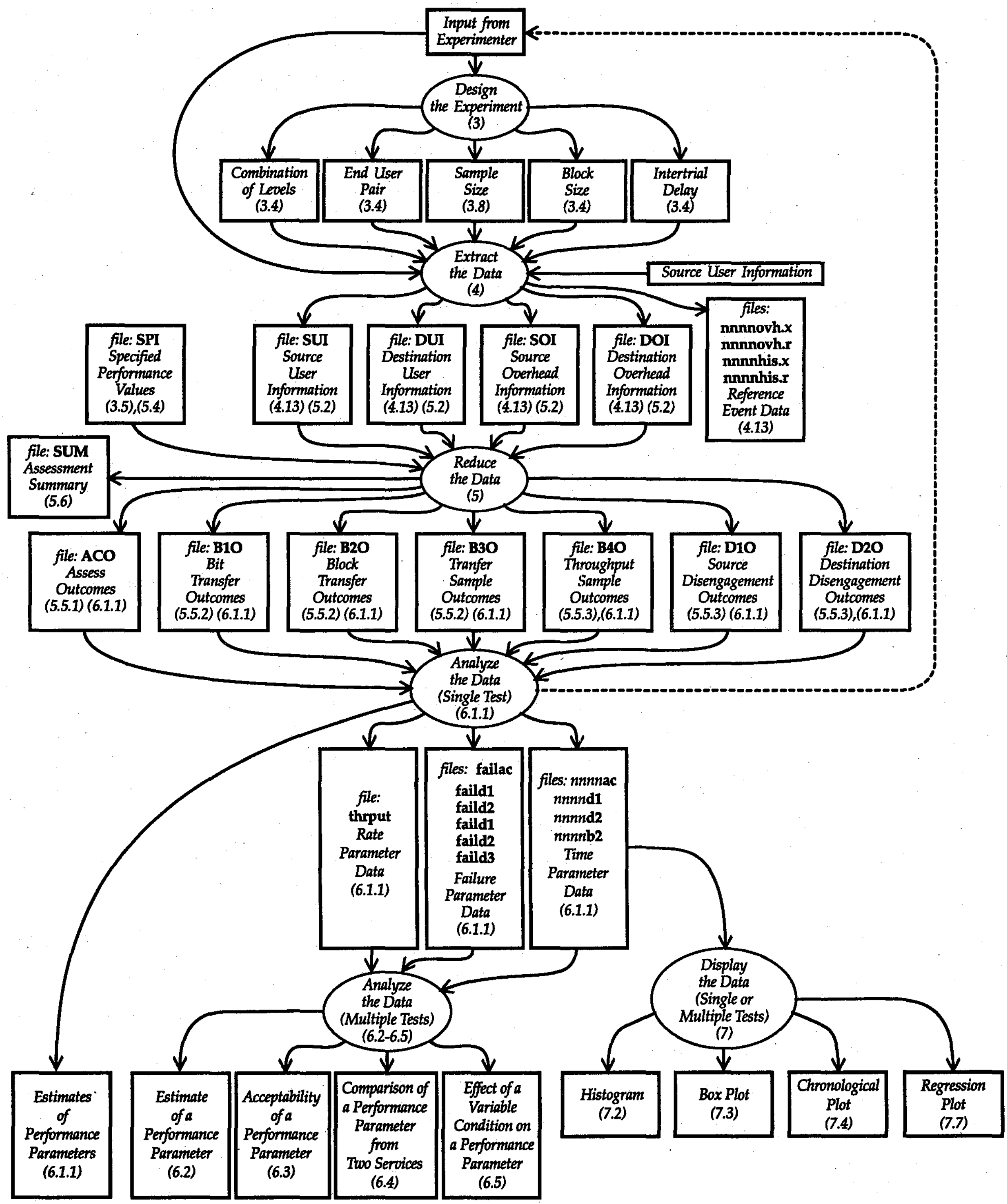


Figure 19. Structured design diagram of the NTIA design of ANS X3.141 specifications.

Section 7). Multiple tests culminate with data analysis (described in Sections 6.2-6.5) and data display (described in Section 7).

It might be helpful to note that each section in Part II in this overview corresponds to the corresponding section in the appropriate volume if the first digit (and its following decimal) is omitted. For example, the section labelled "3.2.1 Recommended Analysis" in this overview corresponds to the section labelled "2.1 Recommended Analysis" in Volume 2.

Where appropriate, structured design diagrams of each phase are shown in the sections that describe them. Each major procedure is assigned an ordered pair of numbers that corresponds to the volume and section, respectively, in which it is discussed; the structured design diagrams in this volume describe the experimenter/operator procedures only. The more detailed structured design diagrams of software procedures are shown in their appropriate volumes.

3. DESIGN THE EXPERIMENT

3.1 Introduction

The design of an experiment should be the first step in conducting an experiment, yet it is the step most frequently neglected. To indicate the importance of experiment design, Table 10 is a list of decisions that must be addressed before any measurements are obtained.

Inaccuracy of an estimate is the bias plus the imprecision. The following four general principals of experiment design minimize the inaccuracy of an estimate:[39]

- Randomization. Systematic errors cause bias. Randomization eliminates systematic errors, thus providing an unbiased estimate.

- Replication. Random errors cause imprecision. Replication reduces imprecision. Precision is usually measured by the length of a confidence interval about the estimate for a specified confidence level (e.g., 95%). If trials are independent, imprecision decreases as $1/\sqrt{n}$ where n is the number of trials. However, imprecision is usually not decreased that much if trials are dependent - a condition that can exist if consecutive trials occur too close in time and space.

- Blocking. Precision is increased by grouping trials measured at multiple levels of a condition that produces similar results. (Grouping trials from multiple levels is equivalent to increasing replication at one of those levels.)

- Balance. Balance is achieved by obtaining the same number of trials in each test. This tends to achieve the same precision for each test - which, in turn, maximizes the precision of a comparison of, say, two performance parameters. Balance also helps determine the relative effects of different conditions upon the performance parameter.

The NTIA design considers dependence between consecutive trials (by using a first-order Markov chain model). Considering dependence does not remove the bias of an estimate, but it does improve its precision (Crow and Miles, 1977).

[39]The first principal tends to reduce bias, and the other three principals tend to reduce imprecision.

Table 10. Decisions Required to Design an Experiment of Data Communication Services

• Determine the objective of the experiment.*
• Select the analysis to be performed.
• Select the most important performance parameter from the access-disengagement function.
• Select the most important performance parameter from the user information transfer function.
• Identify the fixed conditions that exist during the experiment.
• Identify the variable conditions and their levels that exist during the experiment.**
• Specify the performance values.
• Select the single most representative combination of levels of variable conditions for each type of test.
• Select the block size and number of blocks.
• Specify the intertrial delay for each type of test.
• Select the confidence level.
• Select combinations of levels of the variable conditions.
• Specify the desired precision.
• Estimate the dispersion of the most important time performance parameters.***
• Estimate the autocorrelation of lag 1 of the most important time performance parameters.
• Estimate the conditional probability (of a failure given that a failure occurred on the previous trial) of the most important failure probability performance parameter.

*Even though ANS X3.141 states that determining the objective is part of the experiment design phase, it seems that it would be known before the experimenter decides to implement the standard.

**The word "factor" is statistical parlance to describe a variable condition having more than one level (or value). In this report a variable condition is called a factor only after it has been shown to significantly influence the trials of a test.

***This and the following two estimates are obtained if the recommended preliminary characterization test is conducted.

Dependence among trials in a series of trials is called autocorrelation. It is measured by a function called the autocorrelation coefficient that has values between +1 and -1. Dependence usually causes positive autocorrelation. When it does, the confidence interval for a given number of trials is widened. Hence, when positive autocorrelation exists, more trials are required to achieve a specified precision (i.e., width).

Figure 20 is a structured design diagram relating the major procedures in experiment design. It is separated into two diagrams, one for acceptance/maintenance objectives and one for other objectives. Each bubble lists the number of the section in which the procedure is discussed. The following sections describe these procedures.

3.2 Determine the Objective of the Experiment and Select its Analysis

The two principals that are interested in the results of the experiment are usually the user and the vendor. The type of analysis to be selected depends upon the objectives of the experiment. The recommended analyses are introduced here, and the common experiment objectives are discussed in Section 3.2.2.

3.2.1 Recommended Analyses

Although a great variety of analyses are possible, ANS X3.141 recommends one or more of four methods. These methods are discussed with the discussion of the analysis phase (Section 6), but they are introduced here to enable the experimenter to understand the implications of the analyses upon the design of the experiment.

A. Estimation

A performance parameter and its confidence limits can be estimated from either data from a single test (conducted under a single combination of levels of variable conditions) or pooled data from multiple tests (each conducted under different combinations of levels of variable conditions). By virtue of the larger number of trials and the greater variety of conditions, pooled data usually provide a more precise estimate than data from a single test.

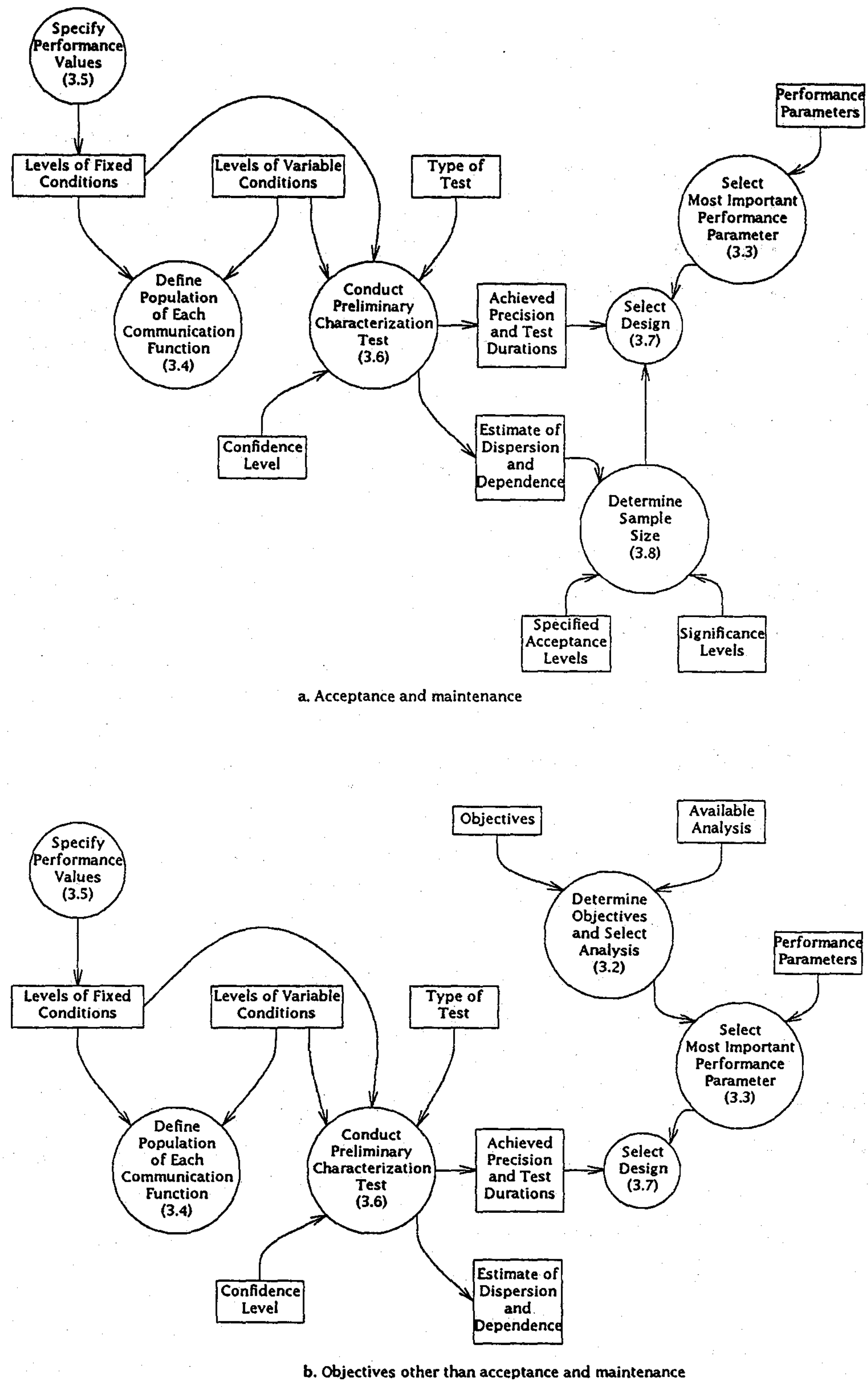


Figure 20. Structured design diagram of the NTIA design of experiment design.

B. Acceptance Test

The null hypothesis for an acceptance test states that the performance parameter mean (or proportion) is equal to an acceptable (threshold) value. An acceptance test determines whether this hypothesis is true. Because of sampling error, an interval of uncertainty exists about the acceptable value. The precision of the test is defined in terms of the length of the interval and the probability of making an incorrect decision when the performance parameter value falls outside that interval. An acceptance test uses data from a single test.

C. Comparison Test

The null hypothesis for a comparison test states that two performance parameter means (or proportions) are equal. The comparison test determines whether the value of a performance parameter from one service is significantly different from that of another. If there is a significant difference, the experimenter can identify the superior service. This test is a special case of the following test.

D. Test to Determine if a Variable Condition is a Factor

A population is defined by a number of fixed conditions and a number of variable conditions; the variable conditions have more than one level (or value). Each test is conducted under a single combination of levels of variable conditions, and multiple tests are usually conducted under a set of combinations of levels of variable conditions. The null hypothesis for these tests states that the means (or proportions) of the multiple populations are equal. A test of this hypothesis can determine whether a variable condition is a factor for a performance parameter. That is, if the null hypothesis is not confirmed by an hypothesis test, the means (or proportions) are concluded to be not equal, and one or more variable conditions are concluded to be factors.

Two types of designs can be used to determine whether variable conditions are factors: complete randomization and randomized blocks. Complete randomization is the simpler design because it requires only that the tests (each at different combinations of levels) occur in a random order. There are many types of randomized block designs. They range from one that determines if a single variable condition is a factor to one that determines if all tested variable conditions are factors.[40]

3.2.2 Common Experiment Objectives

There are six rather common objectives of an experiment of a data communication service. The experimenter should also specify a criterion or criteria to determine whether or not the objective is achieved by the experiment.

A. Acceptance

An experiment for acceptance will be required by the prospective user and, perhaps, by the vendor. It should be conducted only under the conditions and levels expected to be typical. That is, the user's facilities and activities are regarded as fixed; hence, the experiment is usually performed "out of service" for a finite duration to determine whether long-term values of performance parameters lie between specified values.

B. Characterization

A characterization experiment is conducted to estimate a performance parameter. Both the performance parameter and its confidence limits are estimated. The data can come from a single test or from multiple tests. In the latter case, the data are pooled and the resulting estimate may be representative of a range of levels.

C. Maintenance

A maintenance experiment can be conducted by the vendor or the user under a single combination of levels of conditions - a combination of levels that is typical of the user's facilities and activities.

The need for repair can be determined by an acceptance test (or by a quality control chart if values are obtained during operation). The null hypothesis of two equal means is tested. One mean is estimated from current

---

[40]For even as few as three variable conditions, this design can be impractical to implement.

values and the other mean is a specified value (a value, perhaps, guaranteed by the vendor as a result of a characterization test).

D. Design and Management

Design and management are objectives that could include many activities and goals. Depending upon these, the experiment to design and manage data communication services can probably be treated by any or all of the analyses.

E. Optimization

The goal of this experiment is to identify the optimum level of each variable condition upon the performance parameter, but not necessarily independent of the levels of other variable conditions. For example, an optimization experiment would permit the selection of block length to optimize the User Information Bit Transfer Rate (i.e., throughput). Tests for optimization are conducted under a combination of levels of conditions. Experiments for this objective can be conducted by either the vendor or the user.

F. Selection

The selection experiment is usually conducted by the user to choose between two or more services. Tests in this experiment are conducted under a selected combination of levels of variable conditions - the same combination for each service. The null hypothesis of equal means (from the two services) is tested.

3.2.3 Select the Analyses to Meet the Objectives

The analyses and objectives discussed in this section should not be considered to be exhaustive. The objectives are common objectives and the analyses plausible. Table 11 matches these objectives and analyses. The "√" indicates the analysis appropriate for each objective. After the analysis is selected, one can judiciously choose the number of levels of the variable conditions.

Table 11. Common Experiment Objectives and Plausible Analyses

	PLAUSIBLE ANALYSES			
EXPERIMENT OBJECTIVE	Estimation	Acceptance	Comparison	Factor
Acceptance/Maintenance		√		
Characterization	√			
Design/Management	√	√	√	√
Optimization				√
Selection			√	

3.3 Select the Most Important Performance Parameters

The primary data communication functions are access, user information transfer, and disengagement. The first step in selecting performance parameters is to determine which primary data communication function is most important to the objective.[41]

The NTIA design provides two types of measurement tests: Tests of access-disengagement performance parameters and tests of user information transfer performance parameters. Select the most important performance parameter from each type of test.[42]

3.4 Define the Population of Each Communication Function

Each data communication function is defined by a population whose characteristics are its performance parameters. The set of all attempts to complete each function is the population. For example, each access attempt is a trial of the hypothetical (i.e., infinitely large) population of access attempts.

---

[41]The performance parameters are listed in Table 4 according to the data communication functions. Bit Misdelivery and Block Misdelivery are optional performance parameters that have not been implemented by NTIA.

[42]User Information Bit Transfer Rate and User Fraction of Input/Output Time have at most one trial per test. Neither should be selected as the most important parameter because their selection would not provide precision.

The populations are defined by the conditions existing during the experiment – both fixed conditions and variable conditions. A fixed condition will either have no options or the experimenter will choose a single option (a single level, such as a model or a feature) for the entire experiment. Variable conditions have more than one level – and, hence, present an opportunity for experiment design; <u>the experiment is conducted over selected combinations of levels of the variable conditions</u>.

3.5 Specify Performance Values

Before the experiment is conducted, some performance values must be specified to declare failures (and to assign responsibility for them) and to identify outcomes of bits from user information transfer.

An attempt to perform a primary data communication function may be declared a failure if it takes "too long." For example, after a delay that exceeds a factor of three times the specified delay (called a timeout), an access attempt would not result in the Successful Access outcome, but, perhaps the Access Denial outcome if a factor of one times the specified User Fraction of Access Time is not exceeded.

A transfer availability trial is declared a failure if the "quality" of transferred user information is worse than the "quality" of any of four threshold values. The threshold values are functions of values specified for four other performance parameters – three that measure the probability of failures occurring in the bits of the transfer availability trial and one that measures the rate of their transfer. The outcomes of the transfer availability trials during a test define the performance parameter, Transfer Denial Probability.

Other values must be specified to identify outcomes of bits from user information transfer. Sequences of source and destination user information bits are compared to identify bit outcomes. The speed and accuracy of identification depends upon the values specified for comparison.

3.6 Conduct a Preliminary Characterization Test and Determine the Test Duration

Since the values of the performance parameters are generally not known – possibly not even within an order of magnitude – it is prudent or necessary to conduct a preliminary test to characterize the service.[43] For each performance parameter, the preliminary characterization test provides the following:

- Estimates of Statistics. It provides the estimates of some statistics required to estimate the distribution and dependence.

- Set of Precisions. It provides a set of precisions from which one is to be selected. To aid the selection, it lists

  - the number of trials required to achieve that precision, and

  - the test duration required to achieve that precision.[44]

The experimenter specifies a precision or test duration and selects a sample size that will best achieve the specified precision or duration to match the budget.

3.7 Select the Design

The design must also be selected to meet the specified precision, time, and budget. Following are four common designs:

- Factorial Design. A (full) factorial design requires that a test be conducted for every combination of selected levels of variable conditions. This is the most desirable design, but it is the most time consuming.

- Fractional Factorial Design. This design is similar to the factorial design, but with some combinations of levels omitted.

- Blocking Design. An extraneous variable condition is one whose effects are not of interest; it is considered to be a "nuisance" variable condition. Levels of extraneous variable conditions (such as times of day) can be combined either randomly or systematically to remove their effect.

- Response Surface Design. A primary variable condition is one whose effect is of interest; it is thought to have a direct

---

[43]Moreover, this test may reveal that some specified performance values are inappropriate.

[44]The number of trials (or observations) in a test is the sample size. The precision of an estimate usually increases with the sample size.

effect upon a performance parameter. If levels of primary variable conditions are quantitative, orthogonal composite designs can be used, and a regression surface can be fitted.

3.8 Determine the Sample Size

The values obtained from a preliminary characterization test are very helpful to accurately estimate the sample size required to achieve a specified precision. For most objectives, the sample size is obtained from the preliminary characterization test. (It can also be obtained from an interactive program called **star**.[45])

However, for the acceptance objective, the sample size is determined from an operating characteristic curve. Because a sample has a finite number of trials, an interval of uncertainty exists about an acceptable (threshold) level. Acceptance tests attempt to avoid two errors with stated precision:

- Avoid a Type I Error. The probability of rejecting a performance value which is totally satisfactory is to be, say, α = 5% or less.
- Avoid a Type II Error. The probability of accepting the performance parameter when its value is totally unsatisfactory is β. This probability is reduced by selecting a sufficiently large sample size.

[45]**star** is a FORTRAN program that accomplishes three primary tasks; it determines the sample size required to estimate a performance parameter with a specified precision (documented in appendices A and C of Volume 2); it estimates a performance parameter from a single test with stated precision (documented in appendices A, B, C, and D of Volume 5); and it analyzes a performance parameter from multiple tests (documented in appendices E, F, G, and H of Volume 5).

4. EXTRACT THE DATA

4.1 Introduction

Extraction of the data is the most system-specific phase of the NTIA design. It requires hardware and software to access and disengage, to terminate connections, transfer user information, and record interface events at the user-system interfaces.

Figure 21 is a schematic diagram that shows the first four principal elements of the data communication process (Section 1.2): the data communication system, end users (i.e., the application programs, **xmit_net** and **recv**), user-system interfaces (i.e., functional interfaces), and interface events. It also shows satellite clock receivers used to time interface events at the two interfaces.

Each computer contains an application program that performs all local user and interface monitoring activities. The originating end user initiates the test. The source user generates the user information to be transferred. The data extraction software is written in C language. It performs on-line and off-line activities:

- On-Line Software. The on-line data extraction software establishes and terminates connections, transfers user information, and records performance-significant interface events.

- Off-Line Software. The off-line data extraction software consolidates, merges, and reformats the user data (from binary to ASCII text). It also reduces and analyzes the extracted data from each test.

Figure 22 is a structured design diagram that shows the NTIA design of data extraction. Each bubble lists the number of the section in which the procedure is discussed.

4.2 Determine the Protocol

For the NTIA design, a protocol is a sequence of commands and expected responses required to complete the functions of access and disengagement. The interface events from the selected protocol can be determined from protocol analyzers, manual operation, user's manuals, and experts.

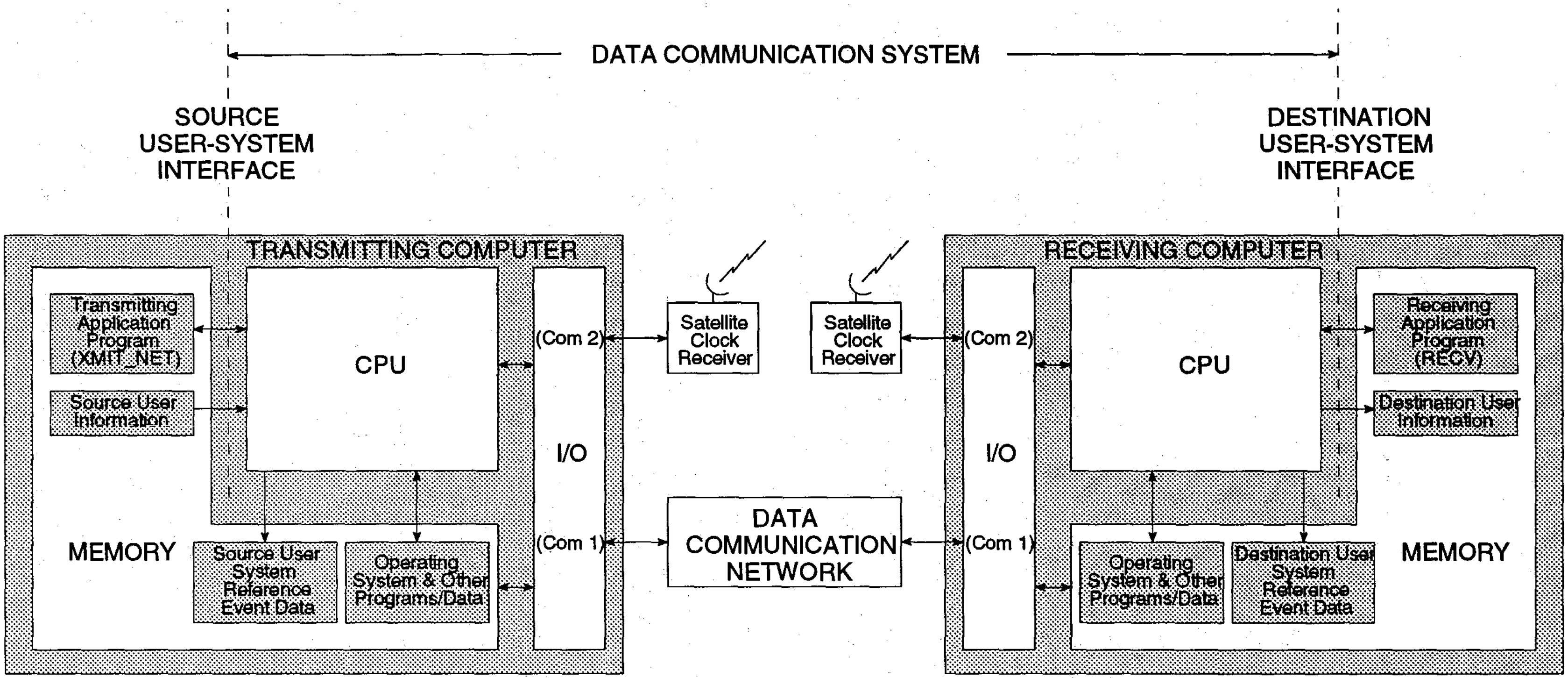


Figure 21. Schematic diagram of a data communication system and end users.

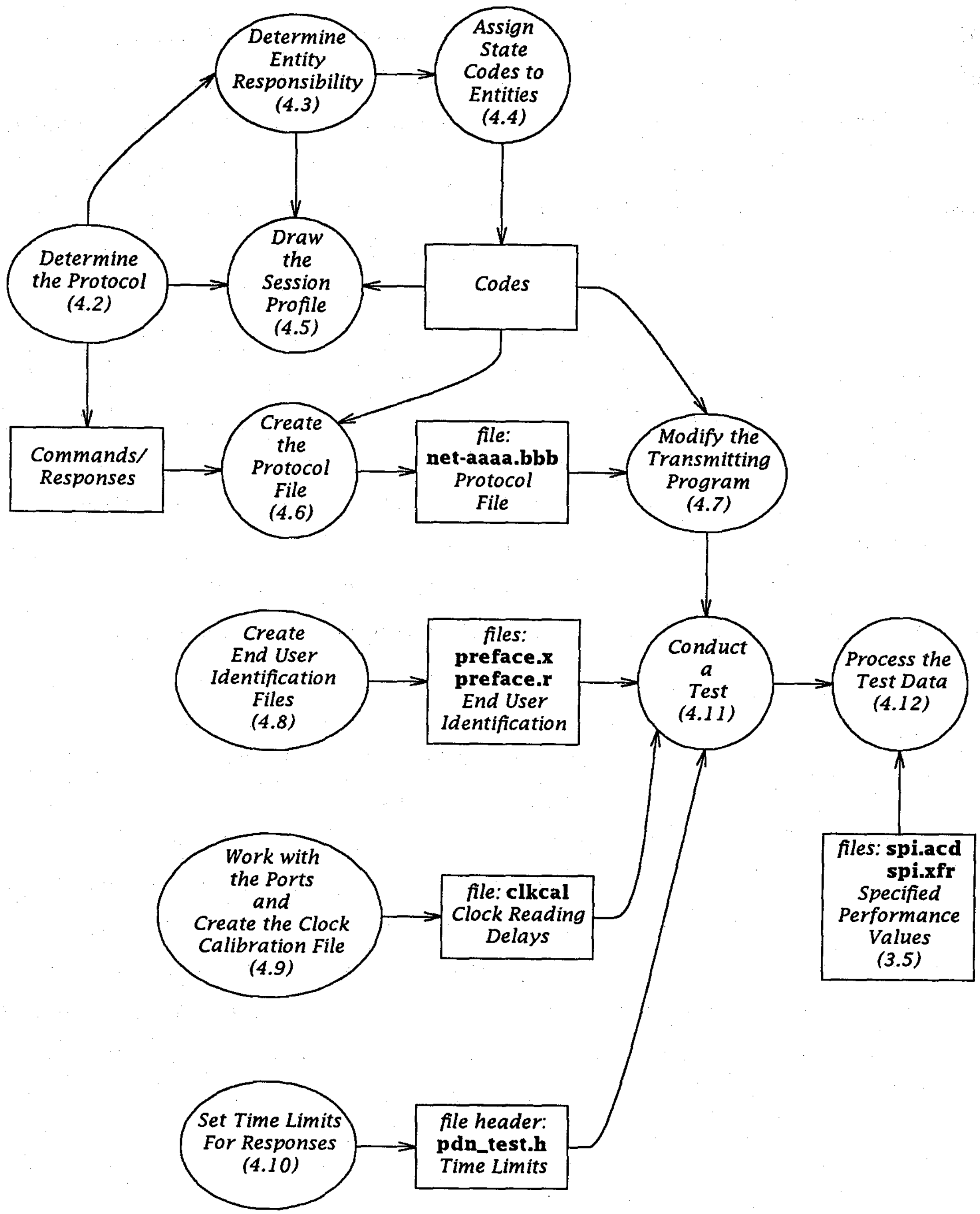


Figure 22. Structured design diagram of NTIA operator implementation of data extraction.

Consider a schematic protocol for a connection-oriented session. It contains five nonspecified commands issued by the originating end user (i.e., C_1, ..., C_5) and their nonspecified expected responses (i.e., R_1, ..., R_5). Table 12 lists the primary data communication states (discussed briefly in Section 4.4), primary data communication functions, the primary reference events, and the symbols for the selected commands and their expected responses. The primary reference events that result from blocking are not identified by a symbol. This schematic protocol will be used in Sections 4.5 and 4.6 to demonstrate a session profile and its corresponding protocol file.

4.3 Determine Entity Responsibility

In the NTIA design of ANS X3.141, the participating entities are the source end user (the application program, **xmit_net**), the system, and the destination end user (the application program, **recv**). The three entities define two interfaces: a source user-system interface and a destination user-system interface.

After a reference event occurs at an interface, usually one or more entities are responsible for producing the next reference event at a given interface. Responsibility must be determined so state codes can be assigned. Responsibility is determined as in Section 2.3.

4.4 Assign State Codes to Entities

An entity can exist in one of three primary states: Idle, Access, and Disengagement. (Although user information transfer is a separate primary state, it is combined with access for the purpose of assigning state codes - because user information transfer doesn't require a protocol.)

NTIA software requires each entity to have a code number that identifies its primary state and its secondary state (i.e., its responsibility for producing the next reference event at a given interface). State codes can be examined and used to determine which events have occurred.

Table 12. Commands and Expected Responses and Their Corresponding Primary Reference Events

PRIMARY DATA COMMUNICATION STATES/FUNCTIONS	PRIMARY REFERENCE EVENTS	COMMANDS/ EXPECTED RESPONSES
(Initial) Idle	-	C_1
	-	R_1
Access	Access Request	C_2
	-	R_2
	Nonoriginating User Commitment	C_3
	System Blocking Signal	-
	User Blocking Signal	-
	-	R_3
	Access Confirmation	(read Block j)
User Information Transfer	Start of Block Transfer	(read Block j)
	End of Block Transfer	(ready)
Disengagement	Source Disengagement Request	C_4
	Source User Disengagement Blocking Signal	-
	Destination Disengagement Request	(ready)
	Destination User Disengagement Blocking Signal	-
	Destination Disengagement Confirmation	R_4
	-	C_5
	Source Disengagement Confirmation	R_5
(Final) Idle	-	-

4.5 Draw the Session Profile

A session profile depicts a data communication session by showing the coordinated sequence of end user and system activities intended to transfer user information. This section implies how a session profile could be drawn by discussing a schematic session profile that includes the seven major concepts of a data communication session (as discussed in Section 1.2) as well as state codes.

Figure 23 is a schematic session profile depicting a normal connection-oriented session involving two end users in which user information is transferred in one direction. Moreover, it corresponds with the schematic protocol file whose commands and expected responses are listed in Table 12. The NTIA implementation of the standard utilizes application programs that have been read into user program memory at the source and destination locations, and user information that has been read into user data memory at the source and destination locations.

4.5.1 Participating Entity Conventions

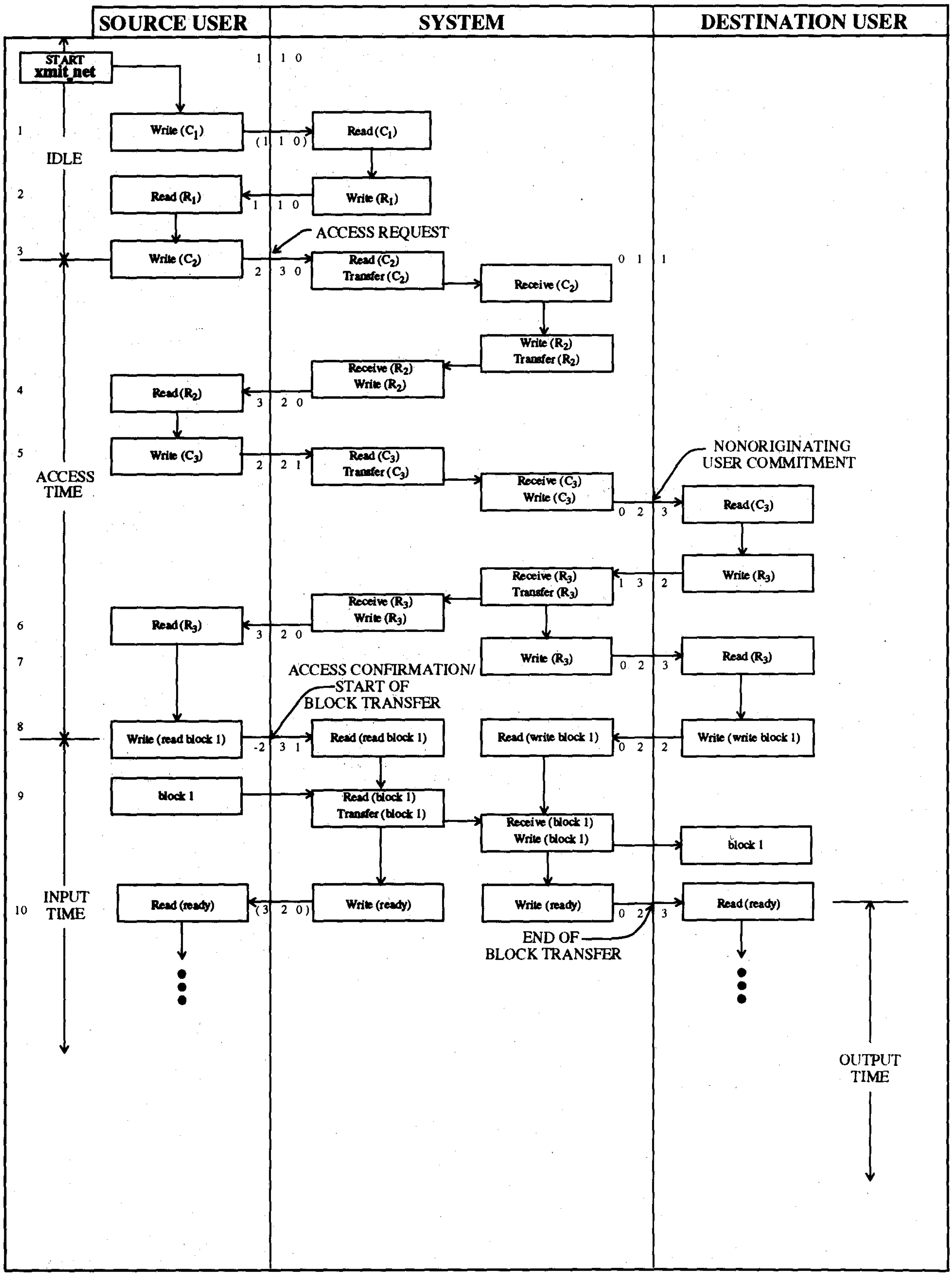
Before discussing the schematic session profile, the following portrayal of the participating entities and their operations are described.

A. Active Peer Entities

The participating entities are the end users and the system. End users are active in one sense: They provide logic (i.e., application program instructions cause the system to perform operations). Systems are active in two senses: They provide logic (i.e., operating system instructions cause the system to perform operations), and they provide the (electric and magnetic) means to execute instructions from both the end user application programs and their operating systems. Because logic is provided by both entities, the schematic session profile portrays both the end users and the system as active peer entities.

B. Entity Operations

Four types of entity operations are specified in the schematic session profile: read, write, transfer, and receive. Each operation has an operand that


SOURCE USER
SYSTEM
DESTINATION USER
START
xmit_net
IDLE
ACCESS TIME
INPUT TIME
OUTPUT TIME
Write (C1)
Read (C1)
Read (R1)
Write (R1)
ACCESS REQUEST
Write (C2)
Read (C2)
Transfer (C2)
Receive (C2)
Write (R2)
Transfer (R2)
Receive (R2)
Write (R2)
Read (R2)
Write (C3)
Read (C3)
Transfer (C3)
Receive (C3)
Write (C3)
NONORIGINATING
USER COMMITMENT
Read (C3)
Write (R3)
Receive (R3)
Transfer (R3)
Receive (R3)
Write (R3)
Read (R3)
Write (R3)
Read (R3)
ACCESS CONFIRMATION/
START OF
BLOCK TRANSFER
Write (read block 1)
Read (read block 1)
Read (write block 1)
Write (write block 1)
block 1
Read (block 1)
Transfer (block 1)
Receive (block 1)
Write (block 1)
block 1
Read (ready)
Write (ready)
Write (ready)
Read (ready)
END OF
BLOCK TRANSFER

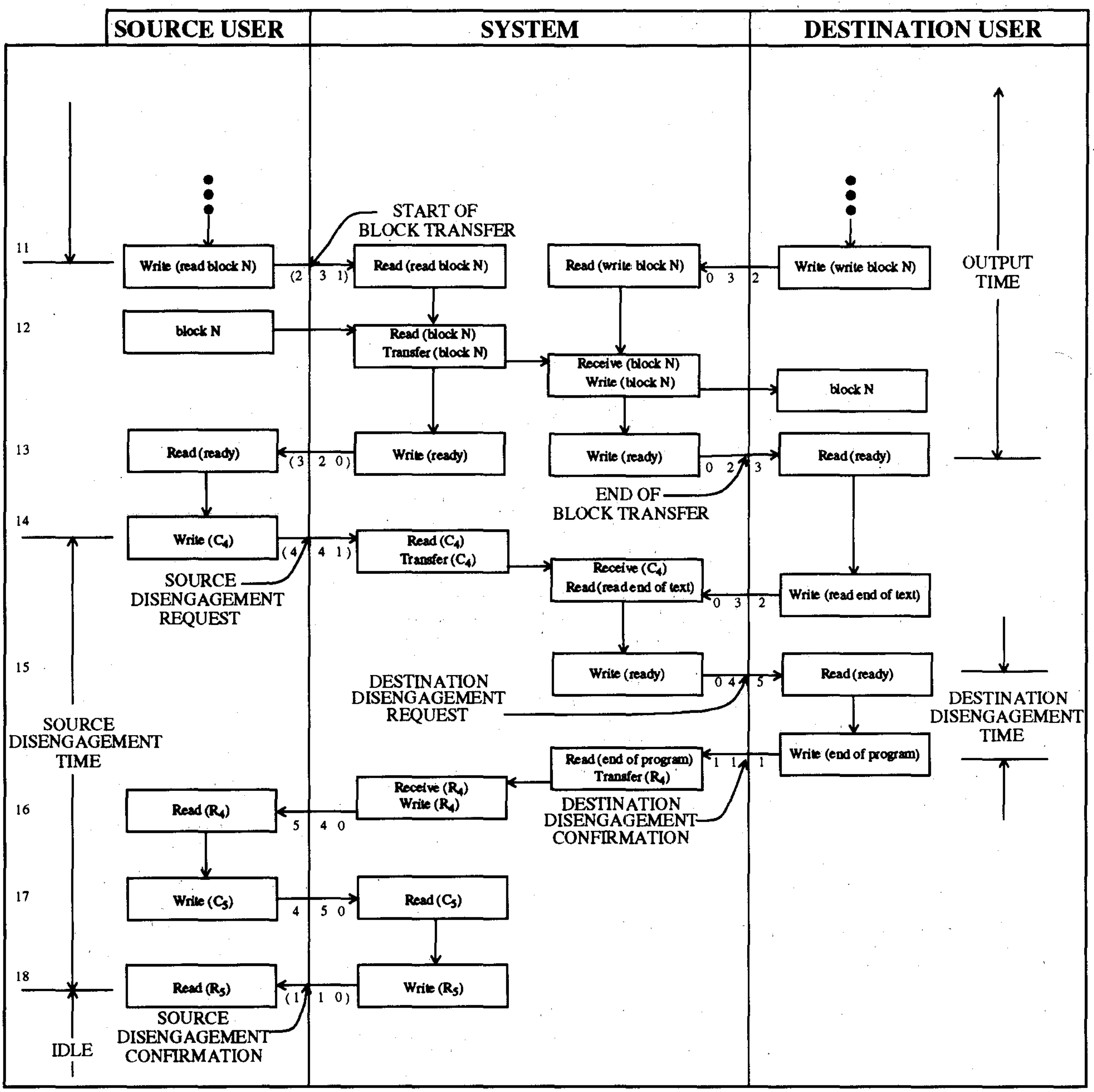


Figure 23. Session profile for the schematic protocol.

is either an instruction or user information. The read and write operations represent user-system interface events as operands that are passed between end users and the proximal portions of the system. However, the transfer and receive operations do not create user-system interface events; they are activities within the system.

A data communication session may require thousands of instructions to be read or written. Moreover, several steps (i.e., interface events) are required to complete each of them.[46] However, the schematic session profile shows only those user-system interface events that are necessary to evaluate performance (except blocking events).

Since the schematic session profile is intended to be expository, it states the fact that all instructions are read or written - even if the instruction, itself, causes the system to read or write. For example, the symbolic instruction, "read block 1", which causes the system to read block 1, must first be written by the end user to the system. Hence, this write operation has the instruction, "read block 1", as its operand, and it is listed in the schematic session profile as

Write (read block 1).

4.5.2 Schematic Session Profile

The schematic session profile shows all seven major concepts of the data communication process (Section 1.2) as well as state codes:

A. End Users

End users are considered to be application programs in user program memory <u>and</u> user information in user data memory:

- <u>Application Program</u>. The application program is a sequence of instructions in user program memory. The instructions consist of commands and expected responses that are symbolized in the

---

[46]As CPUs are currently designed, each instruction must be fetched (from program memory), decoded (by the instruction decoding unit), and executed (by the timing and control unit). Fetching requires a few program memory-CPU interface events, and execution requires a few data memory-CPU interface events (if the instruction has data operands).

schematic protocol file as C_1, ..., C_5 and R_1, ... R_5, respectively, and other commands and responses that are symbolized by words within parenthesis.

- User Information. User information is a sequence of blocks in user data memory at the end user locations. It is shown symbolically as Block 1, ..., Block N.

Henceforth, these two end user components are considered to be collocated in memory called "user memory". (This convention is necessary for each user-system interface to be considered a single interface.)

Since end users are considered to be active and peers of the system, they are shown functionally as performing two types of operations:

- Write Operation. End users are said to write application program instructions to the proximal portion of the system (i.e., to the proximal CPU) from user memory.

- Read Operation. End users are said to read system instructions from the proximal portion of the system (i.e., from the proximal CPU) to user memory.

Read and write operations are the two (of four) operations said to be performable by the end user. These operations and their operands (i.e., instructions) are shown in shaded rectangles.

B. Data Communication System

The data communication system includes the operating systems and CPUs at each user-system interface (and much more, of course). The system is shown functionally as performing all four types of operations:

- Read and Write Operations. Application program instructions and user information are read from user memory by the proximal portion of the system, and system instructions are written to user memory by the proximal portion of the system.

- Transfer and Receive Operations. Some application program instructions, some system instructions, and all user information are transferred and received between source and destination portions of the system.

The source and destination portions of the system communicate with each other (via transfer/receive operations), and each portion communicates with its proximal end user (via read/write operations). The four operations and their operands (i.e., instructions or user information) are shown in shaded rectangles. In some cases, the generic response, **ready**, is shown because its meaning depends upon the system protocol. For example, after a block of user information is read from user memory by the system, **ready** could mean that the block has been read, that it has been read and transferred to the destination portion of the system, or that it has been read, transferred to the destination portion of the system, and written to user memory at the destination user.

C. User-System Interfaces

Each user-system interface (existing anywhere on the data bus between the end user memory and the proximal CPU of the system) is a functional interface because the end user is an application program and user information in user memory (i.e., the two components that are considered to be collocated).[47] The two end user-system interfaces are shown as vertical lines separating the two end users and the system.

---

[47]If user information is not considered to be part of the end user, the instructions to read/write it would cross a user-system interface, but the user information would cross a data-system interface - not a user-system interface as required by the standard:

- The end user writes the instruction to read a block.
- The instruction is read by the source portion of the system.
- The source portion of the system reads the block (from data memory - not from user program memory).
- The source portion of the system transfers the block to the destination portion of the system.
- The destination portion of the system receives the block.
- The destination portion of the system writes the block (to data memory - not to user program memory).

D. Interface Events

A data communication session usually requires thousands of instructions to be performed, and each instruction requires several user-system interface events. All user-system interface events that are shown are read/write operations (which symbolize either the completion of an instruction or the reading and writing of user information), and only a few of these need to be shown. Specifically, only a few user-system interface events are primary reference events or ancillary reference events. The locations of all primary reference events, except blocking events, are identified and labelled in the schematic session profile, Figure 23.

E. Transferred Information

There are two types of transferred information:

- User Information. User information is considered to be a component of the end user. The schematic session profile shows the blocks of user information as part of the end user and as being read by the system at the source location and written by the system at the destination location.

- Overhead (Residual) Information. There are three types of overhead information: System control information (which causes user-system interface events - e.g., instructions from an application program), user control information (which also causes user-system interface events), and intra-system coordination information (which does not cause user-system interface events, but utilizes the transfer/receive operations).

F. Data Communication Session

This major concept is formalized by a session profile. In this report, the schematic session profile describes a connection-oriented data communication session between a pair of end users involving user information transfer in one direction. It comprises a single access function, a single user information transfer function, and a disengagement function for each end user. It begins with a user-system interface event that commits the originating user to attempt communication. It includes all subsequent events that depend on that initial event. It ends with an event that renders all subsequent events, involving any user, independent of the original (commitment) event.

G. Primary Data Communication Functions

The schematic session profile shows the primary reference events (except blocking reference events). These reference events are necessary to define the primary data communication functions. Additionally, the profile shows the duration of the primary speed performance parameters and component times of the User Information Bit Transfer Rate: Access Time, Input Time, Output Time, Source Disengagement Time, and Destination Disengagement Time.

H. State Codes

The state codes are not major concepts of the data communication process, but they are shown immediately below each user-system interface event to indicate the state of the entities immediately after the user-system interface event. At the source user interface, some state codes are embedded in the transmitting application program (and shown in Figure 24). At the destination interface, all state codes are embedded in the receiving application program.

4.6 Create the Protocol File

Protocol files are read by the source end user (i.e., the application program, **xmit_net**). One protocol file exists for each data communication system/service (and also for each site that is accessed by a telephone number). It must agree with the session profile at the <u>source</u> user-system interface. This section describes the elements of a protocol file and the schematic protocol file.

4.6.1 Elements of a Protocol File

A protocol file is a text file that contains three types of lines:

- <u>Nonexecutable Comment Lines</u>. These lines have a # in column one and can be placed anywhere.
- <u>Executable Command/Expected Response Lines</u>. These lines contain commands on the left side, the expected response on the right, and the Δ|Δ character sequence between them. The Δ character represents either the blank character or a tab.
- <u>Executable Time Stamp and Entity State Code Lines</u>. These lines cause the time to be recorded. They have an * in column

one. The * is followed by three digits, the Δ|Δ character sequence, and three more digits.

- Left-Hand Three Digits. The left-hand three digits are the entity state codes of the three entity-interface responsibilities about the source interface following the previous response (i.e., the response as shown in the session profile, not as shown in the protocol file): From left-to-right, the digits are

    - source user state code at local interface,
    - system state code at local interface, and
    - system state code at remote interface.

- Right-Hand Three Digits. The right-hand three digits are the entity state codes of the three entity-interface responsibilities about the source interface following the next command (i.e., the command as shown in the session profile, not as shown in the protocol file): From left-to-right, the digits are

    - source user state code at local interface,
    - system state code at local interface, and
    - system state code at remote interface.

Examples of these elements of a protocol file are shown in Figure 24.

4.6.2 Schematic Protocol File

Figure 24 shows the schematic protocol file for a connection-oriented session. It contains the sequence of commands, expected responses, and entity state codes at the source user-system interface. The minus sign preceding the state codes 231 indicates the end of the access state. Only bold characters are part of the protocol file; characters that are not bold (i.e., C_3 =, R_3 =, C_4 =, and R_4 =) are included only to indicate the relationship of these commands and expected responses to those of Table 12 and Figure 23.

```
        C1           Δ|Δ          R1

*       110          Δ|Δ          230

#       Beginning of access

        C2           Δ|Δ          R2

*       320          Δ|Δ          221

    C3 = recv\r      Δ|Δ     R3 = READY

*       320          Δ|Δ         -231

#       End of access - Beginning of user information transfer

#       End of user information transfer - Beginning of disengagement

    C4 = \e          Δ|Δ     R4 = %\s

*       540          Δ|Δ          450

        C5           Δ|Δ          R5

#       End of disengagement
```

Figure 24. Schematic protocol file.

4.7 Modify the Transmitting Program

The transmitting program, **xmit_net**, reads the protocol files and must agree with them. Therefore, this C program must be modified for each protocol.

4.8 Create the End User Identification Files

There is an identification file for each end user (called **preface.x** and **preface.r**). Each file contains seven lines of identification.

4.9 Work with the Ports and Create the Clock Calibration File

Link **com1** and **com2** to the serial ports. Then connect the each computer to the network at **com1** and the satellite clock receiver at **com2**.

The time of each reference event is recorded. However, each recorded time must be corrected due to the transmission delay between the computer at the end user location and the satellite clock receiver. The clock calibration file must include this information.

4.10 Set the Time Limit for Responses

If responses are not read within a specified time, the system may be responsible. A timeout value must be specified so the test isn't suspended indefinitely while awaiting a response. (Hence, this value only needs to be larger than any of the timeout values associated with specified performance values that define primary performance parameter failures.)

4.11 Conduct a Test

At the beginning of the experiment, the UNIX^{tm} clock and the satellite clock must be synchronized. Four **runx** commands are available. They allow the experimenter an option concerning flow control and the adjustment of times for transmission delays (between the host and the satellite clock receiver). Enter a **runx** command with its arguments (between one and three arguments), and move the data collected at the destination site into storage.

4.12 Process the Test Data

The data from the test may be processed by consolidating the extracted data from the two hosts, merging the files, copying the extracted data, and activating a shell script (called **do**) by typing

do nnnn

where **nnnn** is the test number. This command activates a shell script that

- conducts off-line data extraction (called data conversion),
- reduces the data,

- analyzes the data (by producing an estimate of the performance parameters with precision), and
- prepares files for graphical analysis of primary time performance parameters.

4.13 Data Extraction Output

The **do** shell script produces two types of files. The first type consists of four files containing reference event data. They are **nnnnovh.x**, **nnnnovh.r**, **nnnnhis.x**, and **nnnnhis.r**. These files contain information for the experimenter and are not input to other phases. The second type are four files that are input to the data reduction phase. These files are source overhead information (**SOI**), destination overhead information (**DOI**), source user information (**SUI**), and destination user information (**DUI**).[48] These files are identified in Figure 19.

4.13.1 Overhead Files

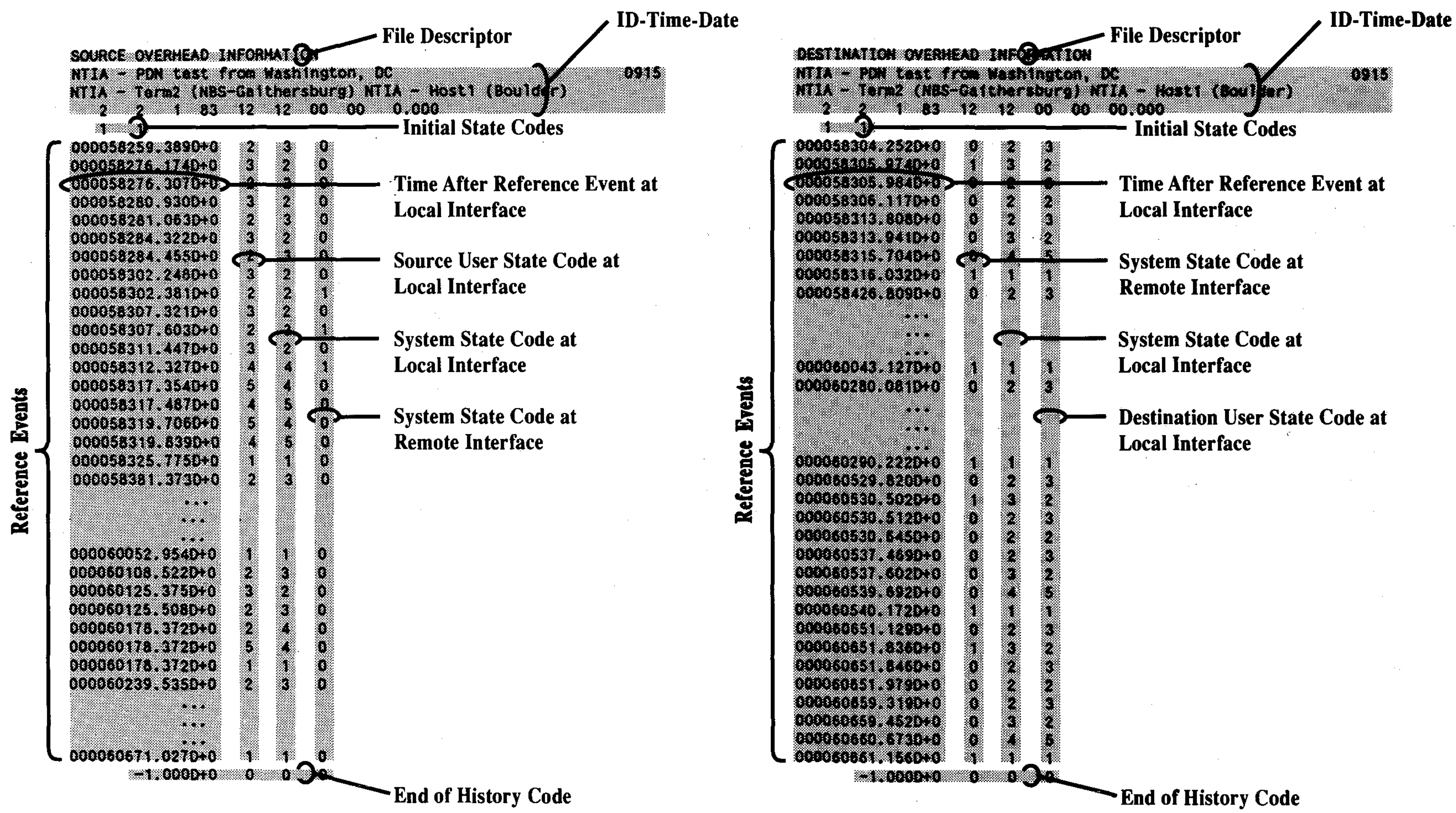
Examples of the two overhead information files are shown in Figure 25.

A. Source Overhead Information

The **SOI** file contains

- a file descriptor,
- three lines of identification, time, and date,
- a line for each reference event containing the time <u>after</u> a reference event at the local interface, the source user state code at the local interface, the system state code at the local interface, the system state code at the remote interface, and
- the end of history codes.

[48]The word "overhead" as used here refers to information associated with an access-disengagement test rather than information that is not user information as defined in ANS X3.102.



a. SOI File

b. DOI File

Figure 25. Example of source (SOI) and destination (DOI) overhead files.

B. Destination Overhead Information

The **DOI** file contains

- a file descriptor,
- three lines of identification, time, and date,
- a line for each reference event containing the time <u>after</u> a reference event at the local interface, the system state code at the remote interface, the system state code at the local interface, and the destination user state code at the local interface, and
- the end of history codes.

4.13.2 User Information Files

Examples of the two user information files are shown in Figure 26.

A. Source User Information

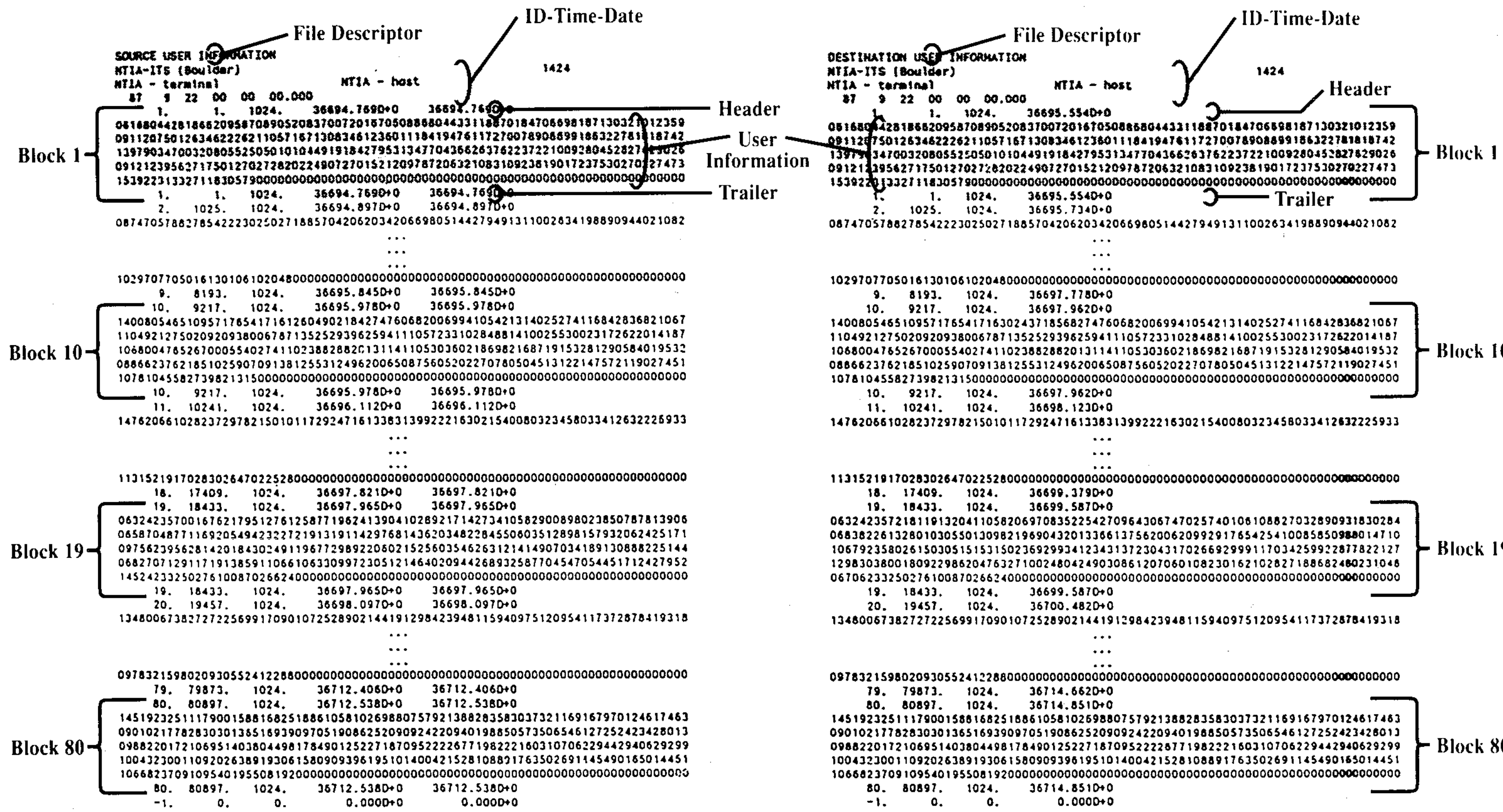
The **SUI** file contains

- a file descriptor,
- batch identifier, source and destination user identification, and date and time of the reference event at the originating user-system interface,
- a header containing the block index, initial bit index, block size, and time of block input after the Start of Block Transfer reference event,
- several lines of 15 characters of source user information in which each character is encoded by a five digit number,
- a trailer having the same information as the header, and
- the end of history, which is a zero or negative number followed by four zeros.

B. Destination User Information

The **DUI** file contains

- a file descriptor,



a. SUI File

b. DUI File

Figure 26. Example of source (SUI) and destination (DUI) user information files.

- batch identifier, source and destination user identification, and date and time of the reference event at the originating user-system interface,
- a header containing the block index, initial bit index, block size, and time after the End of Block Transfer reference event,
- several lines of 15 characters of destination user information in which each character is a five digit number,
- a trailer having the same information as the header, and
- the end of history, which is a zero or negative number followed by four zeros.

5. REDUCE THE DATA

5.1 Introduction

This phase of the NTIA design processes the extracted data for analysis. The principal tasks of data reduction are to identify performance trials, determine their outcomes, and calculate the performance parameters. In other words, this phase is the design of the performance parameter definition process of ANS X3.102 (i.e., shown in Figure 3 and discussed in Section 1.3).

The following section describes the three FORTRAN programs whose executions constitute the data reduction phase. Section 5.3 describes the five input files to this phase. Sections 5.4-5.5 describe the preliminary procedures to examine and consolidate the input data, assess performance, and produce the assessment summary files.

5.2 Synopsis of Data Reduction

The extracted data are reduced when the following three FORTRAN programs are executed in sequence:

- PROLOG. This program examines the input files for format and content, and it consolidates the data from normal tests.
- ANALYZ.[49] This program performs the three principal tasks of data reduction. It identifies individual performance trials, determines their outcomes, and calculates performance parameter values.
- EPILOG. This program produces a user-oriented summary of performance assessment.

Each execution of this sequence of programs is called a reduction run.

5.3 Data Reduction Input

Five ASCII character files are input to data reduction. One file contains the specified performance values (created during the experiment design phase and described in Volume 2), and four files contain performance data (created during the data extraction phase). The four performance data files depend upon whether

[49]Despite the name, this FORTRAN program has nothing to do with the analysis phase of the NTIA implementation.

the end user is the source or destination and whether the test is an access-disengagement (called overhead) test or a user information transfer test. The five ASCII character files are as follows:

- SPI. The specifications file contains a set of specified performance parameter values for determining outcomes of performance trials (e.g., excessive delays are considered to be failures). The files **spi.acd** and **spi.xfr** are created during the experiment design phase, and, depending upon the type of test, the appropriate file is copied into **SPI**.

- SOI. The source overhead information file contains records of identification, times of primary reference events and significant ancillary reference events for one or more sessions, and communication state codes at the source user-system interface

- DOI. The destination overhead information file contains records of identification, times of primary reference events and significant ancillary reference events for one or more sessions, and communications state codes at the destination user-system interface.

- SUI. The source user information file contains records of identification and user information blocks transferred across the source user-system interface during one or more sessions.[50]

- **DUI**. The destination user information file contains records of identification and user information blocks transferred across the destination user-system interface during one or more sessions.

Program **PROLOG** examines the input files for format and content - not for performance.

If errors are not found (in which case the reduction run is called a normal run), the program consolidates the reference event data in the source overhead information (**SOI**) file and the destination overhead information (**DOI**) file to

---

[50]Each group of 15 user information bits is mapped into ASCII character representation (i.e., a five-digit decimal integer) in the range

$$[0, 2^{15} - 1] = [0, 32{,}767].$$

The data are read and converted by FORTRAN routines using an I5 edit descriptor. Hence, the data reduction software can run on computers that allow 16-bit locations for integers; user information can be stored in the sign bit of these locations.

create the consolidated overhead information (COI) file, and it combines the specification (SPI) file with data from these two files to create the consolidated specifications (CSP) file.

If errors in format or content are found (in which case the reduction run is called an exceptional run), it writes a diagnostic to the assessment summary (SUM) file and suspends data reduction for that run.

5.4 Assess Performance

Program **ANALYZ** performs the three principal tasks of data reduction for each of the primary communication functions: access, user information transfer, and disengagement. It

- identifies individual performance trials,
- determines their outcomes, and
- calculates the performance parameter values.

5.4.1 Assess Access Performance

Subroutine **ACCESS** assesses access performance. Specifically, it identifies access attempts, determines access outcomes, and calculates access performance parameters.

A. Identify Access Attempts

Each access attempt

- begins with the primary reference event, Access Request, and
- ends with
  - the first Start of Block Transfer reference event (or the Access Confirmation reference event as defined by the revised version of ANS X3.102),
  - the first System Blocking Signal reference event, or
  - the first User Blocking Signal reference event.

B. Determine Access Outcomes

This subsection is divided into two subsections; the first subsection discusses the outcome scheme, and the second subsection discusses the outcome files.

1. Access Outcome Scheme

Figure 10 is a flowchart showing how the <u>five</u> outcomes of an access attempt are determined according to ANS X3.141. Figure 27 is a similar flowchart showing how the <u>six</u> outcomes of an access attempt are determined according to NTIA. The latter flowchart includes an outcome that is not classified. This outcome can occur from an anomalous blocking event that is attributed to neither the system nor either end user; the end user that issues the User Blocking Signal reference event (whether the originating end user or the nonoriginating end user) is not distinguished in Figure 10 as it is in Figure 27. Outcomes that are not classified are not included in performance measurements.

2. Access Outcome File

An example of the access outcome (**ACO**) file is shown in Figure 28. Each line contains

- the file descriptor,
- the batch identification,
- either the Access Time and the user portion of Access Time for each successful attempt, or a code for each unsuccessful access attempt (i.e., -1, -2, -3, -5, and -9 for Incorrect Access, Access Denial, Access Outage, User Blocking, and nonclassified outcomes, respectively),
- the end of history code (i.e., -30.),
- the number of access attempts and both the number of failures and the number of pairs of consecutive failures for the three types of failures (Incorrect Access, Access Denial, and Access Outage), and
- the specified Access Time and the specified User Fraction of Access Time (both from the **SPI** file).

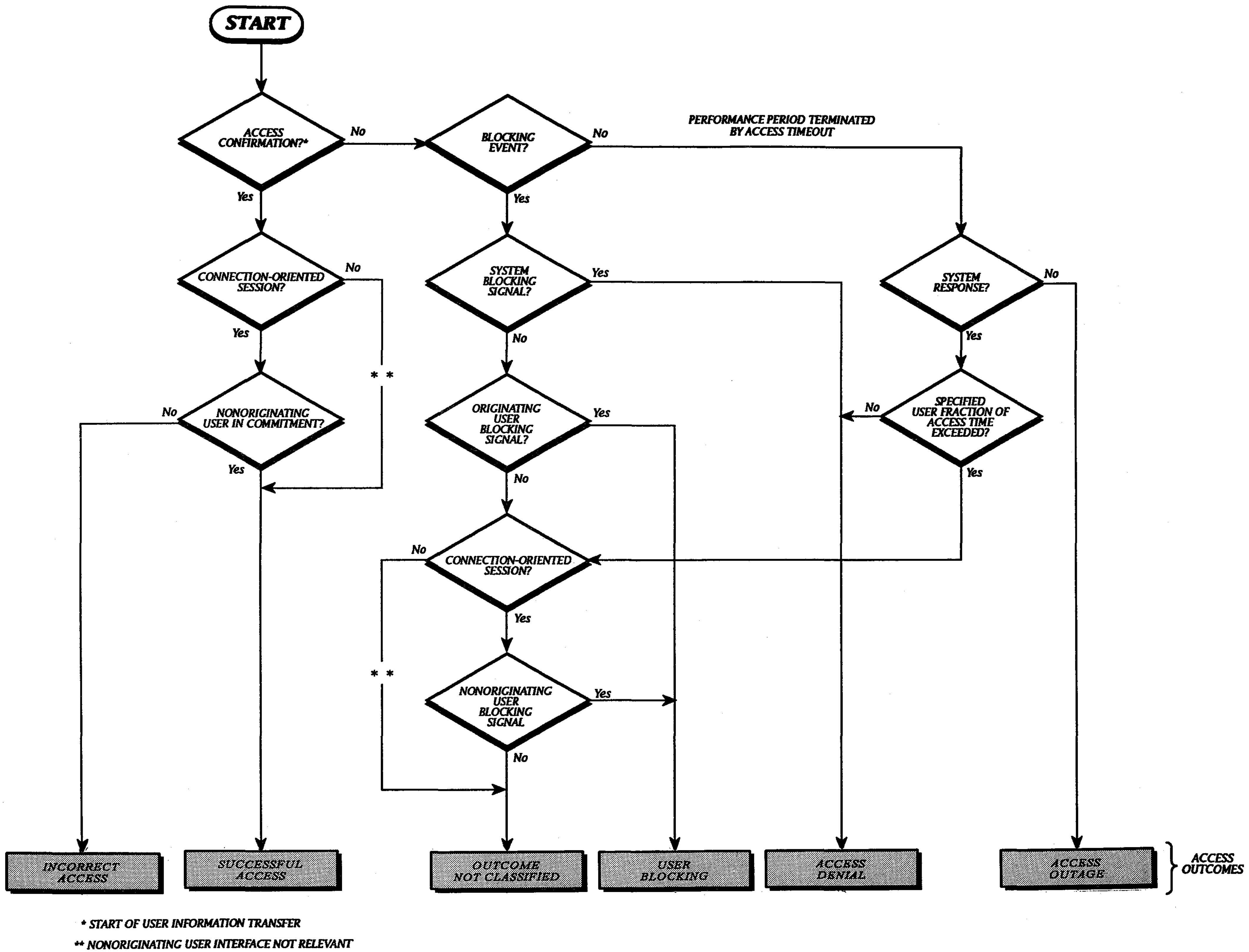


Figure 27. Flowchart that determines access outcomes according to NTIA.

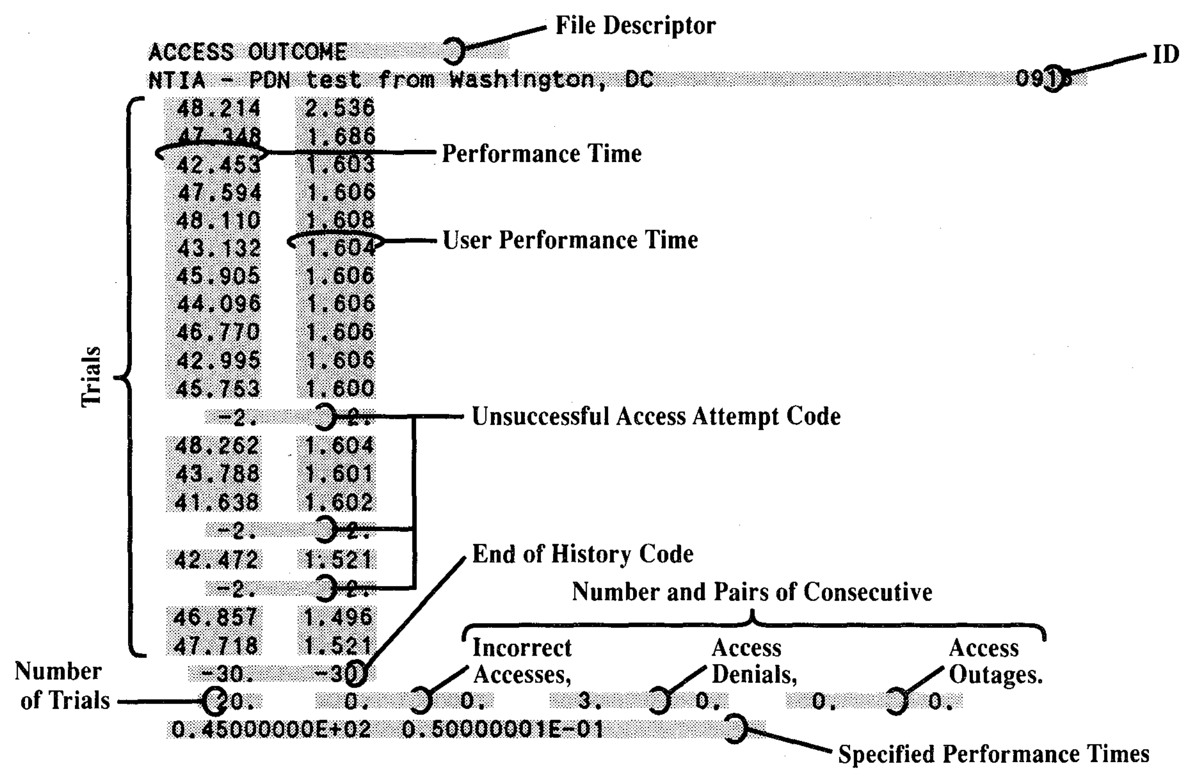


Figure 28. Example of access outcome (ACO) file.

C. Calculate Access Performance Parameters

The five performance parameters from the access function are calculated by the program **ANALYZ** and subroutine **ANCILL** using outcomes in the **ACO** file. They are calculated as they are defined in Section 1.3.1.

5.4.2 Assess User Information Transfer Performance

Subroutine **TRANSF** assesses user information transfer performance. Specifically, it identifies bit and block transfer attempts, determines bit, block, and transfer sample transfer outcomes, and calculates user information performance parameters.

A. Identify User Information Transfer Attempts

TRANSF first calls subroutine **BITCOR** to correlate source and destination user information in the performance batch. Subroutine **BITCOR** identifies bit transfer attempts and block transfer attempts and records them in the correlator output (COR) file. (Because transfer availability trials and the throughput trial are not attempts, they are not identified here.)

B. Determine User Information Transfer Outcomes

This section is divided into two sections; the first section discusses the outcome scheme, and the second section discusses the outcome files.

1. User Information Transfer Outcome Scheme

This section is divided into three sections; the first section discusses the bit and block transfer outcome schemes, the second section discusses the transfer availability trial outcome scheme, and the third section discusses the throughput trial outcome scheme.

a. Bit Transfer and Block Transfer Outcome Schemes

Figure 29 is a flowchart showing how the outcomes of block transfer attempts and bit transfer attempts are determined.[51] It includes bit transfer outcomes and block transfer outcomes that are not classified. The NTIA implementation includes the bit transfer outcomes specified by ANS X3.102 as well as nonclassified outcomes. Nonclassified outcomes may occur when the performance period is terminated abnormally or terminated normally (i.e., when the End of Block Transfer reference event occurs before the Block Transfer Timeout), but neither the destination portion of the system nor the destination user is in the active state.[52]

Table 13 summarizes bit transfer outcomes depending upon block transfer outcomes and Bit Comparison Outcomes (BCOs). Table 14 goes further and indicates by "√" which bit transfer outcomes and block transfer outcomes can coexist; there is a one-to-one correspondence between bit transfer outcomes and block transfer outcomes except

- Incorrect Block outcomes can also contain Successful Bit Transfer outcomes, Extra Bit outcomes, and Lost Bit outcomes, and

- Lost Block outcomes can also contain Extra Bit outcomes.

b. Transfer Availability Trial Outcome Scheme

Transfer availability trials are used to measure Transfer Denial Probability. The NTIA implementation of transfer samples incorporates concepts from both the original and the revised versions of ANS X3.102. The revised version of ANS X3.102 defines a transfer availability trial as the smallest transfer sample that spans the specified minimum Input/Output Time. The original version of ANS X3.102 states that a transfer sample must contain a specified minimum number of bits, and it does not define a transfer availability trial. NTIA has implemented the requirement of the specified minimum number of bits from

---

[51]This flowchart can be compared to the similar flowcharts from ANS X3.141 (Figures 13 and 14).

[52]If an entity is responsible for producing the next reference event and is in either the Committed or the Closing state, it is also in the Active state.

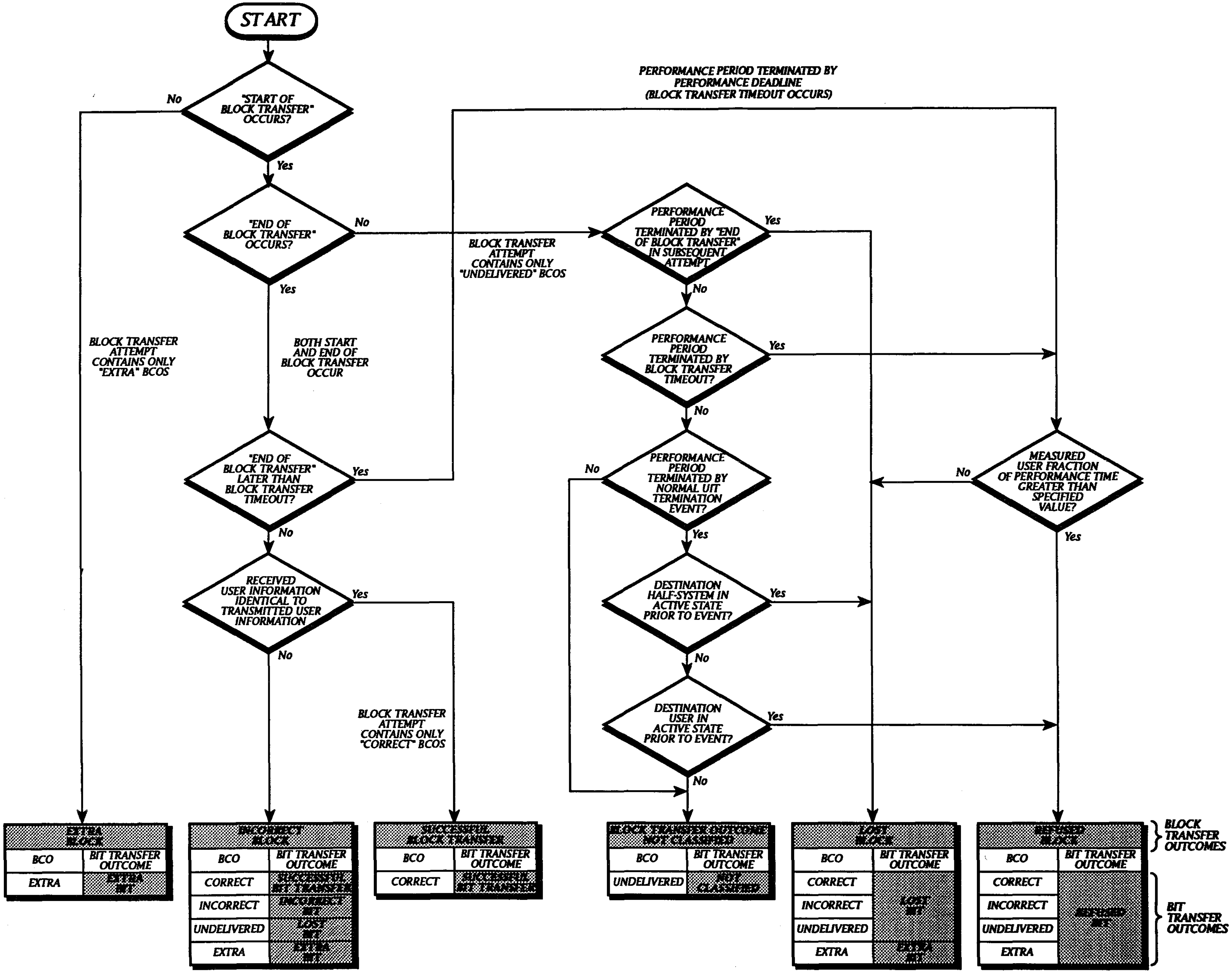


Figure 29. Flowchart that determines bit transfer outcomes and block transfer outcomes according to NTIA.

Table 13. Bit Transfer Outcomes (Except Misdelivered Bit Outcome) Depending upon Bit Comparison Outcomes and Block Transfer Outcomes

	BLOCK TRANSFER OUTCOMES					
BIT COMPARISON OUTCOMES	Successful Block Transfer	Incorrect Block	Extra Block	Lost Block	Refused Block	Block Not Classi-fied
Correct Bit	Successful Bit Transfer	Successful Bit Transfer	-	Lost Bit	Refused Bit	-
Extra Bit	-	Extra Bit	Extra Bit	Extra Bit	Refused Bit	-
Incorrect Bit	-	Incorrect Bit	-	Lost Bit	Refused Bit	-
Undelivered Bit	-	Lost Bit	-	Lost Bit	Refused Bit	Bit Not Classi-fied

Table 14. Bit and Block Transfer Outcomes That Can Coexist

	BLOCK TRANSFER OUTCOMES					
BIT TRANSFER OUTCOMES	Successful Block Transfer	Incorrect Block	Extra Block	Lost Block	Refused Block	Block Not Classi-fied
Successful Bit	√	√	-	-	-	-
Incorrect Bit	-	√	-	-	-	-
Extra Bit	-	√	√	√	-	-
Lost Bit	-	√	-	√	-	-
Refused Bit	-	-	-	-	√	-
Bit Not Classified	-	-	-	-	-	√

the original ANS X3.102, but those transfer samples meeting this requirement (as well as the following requirements) are called transfer availability trials.

Specifically, subroutine **TTSAMP** determines transfer availability trials by the following process:

- Specified Size. The first block transfer attempt must contain a number of bits that is equal to or greater than the minimum number of bits specified for a transfer sample. If it does not, more block transfer attempts are included until the specified number of bits is obtained.

- Requirements for Transfer Sample. The four times (i.e., beginning and ending Input Times and Output Times) of these block transfer attempts and their preceding interblock gaps must qualify as a transfer sample.

- Specified Quality. The outcome of the transfer availability trial is determined according to its quality as in Figure 30. The measures are bit-oriented; specifically, the User Information Bit Transfer Rate is the "short-term" version.

This process is applied to all block transfer attempts in the data communication session. Each transfer availability trial, obtained in this way, is the smallest transfer sample containing the specified minimum number of bits. If the transfer sample contains n block transfer attempts, this process will produce between 0 and n transfer availability trials.

c. Throughput Trial Outcome Scheme

The throughput trial is used to measure the "long-term" version of the User Information Bit Transfer Rate and its ancillary performance parameter, the User Fraction of Input/Output Time. All transfer availability trials in the data communication session are used to determine the throughput trial.[53]

Specifically subroutine **TTSAMP** determines the throughput trial by the following process:

- Specified Size. The throughput trial must be at least as large as a transfer availability trial (i.e., it must contain

[53]Revised ANS X3.102 requires all transfer availability trials used in the throughput trial to have the Successful Transfer outcome.

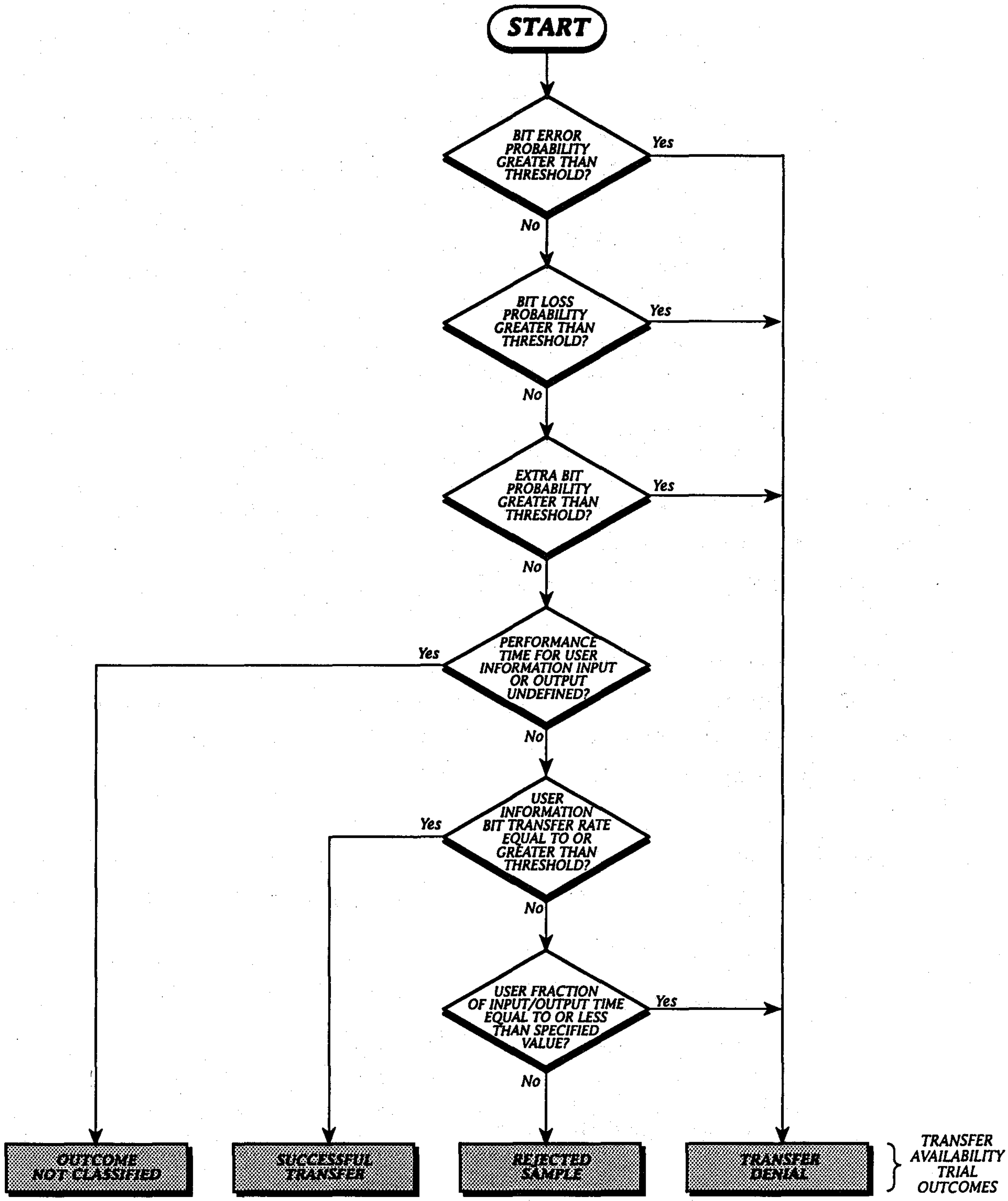


Figure 30. Flowchart that determines transfer availability trial outcomes according to NTIA.

at least the minimum number of bits specified for a transfer sample).

- Requirements for Throughput Trial. The requirements for a throughput trial are those of a transfer availability trial. All four times required of a transfer sample must be obtained. Therefore, a throughput trial is terminated only
    - when **TTSAMP** identifies a block transfer attempt that has neither the Successful Block Transfer outcome nor the Incorrect Block outcome, or
    - when the last block transfer attempt is in the correlator output file (COR).
- Specified Quality. The number of Successful Bit Transfer outcomes in the throughput trial is determined.

Normally there will be one throughput trial for each data communication session, but if the session is very short or the transferred user information is of unusually poor quality, there can be zero throughput trials.

2. User Information Transfer Outcome Files

Four outcome files are produced by the user information transfer software. Subroutine **BITCOR** records bit transfer and block transfer attempts in the correlator output file (COR).

a. Bit Transfer Outcome File

B1O is the bit transfer outcome file (Figure 31a). Each line contains

- the file descriptor,
- the batch identification, and
- the number of bit transfer attempts and both the number of failures and the number of pairs of consecutive failures for the following three types of failure outcomes: Incorrect Bit outcome, Lost Bit outcome, and Extra Bit outcome.

b. Block Transfer Outcome File

B2O is the block transfer outcome file (Figure 31b). Each line contains

- the file descriptor,

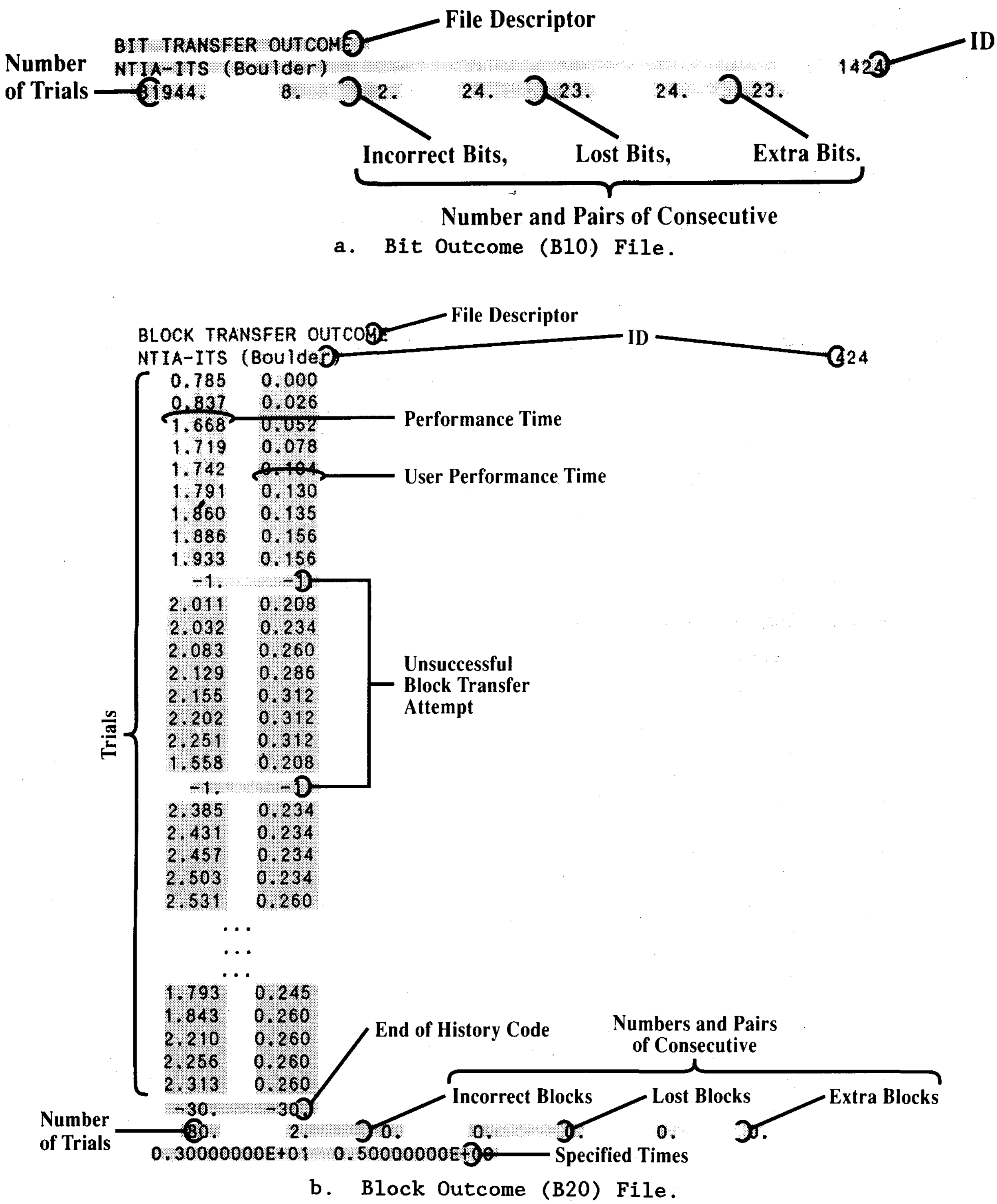


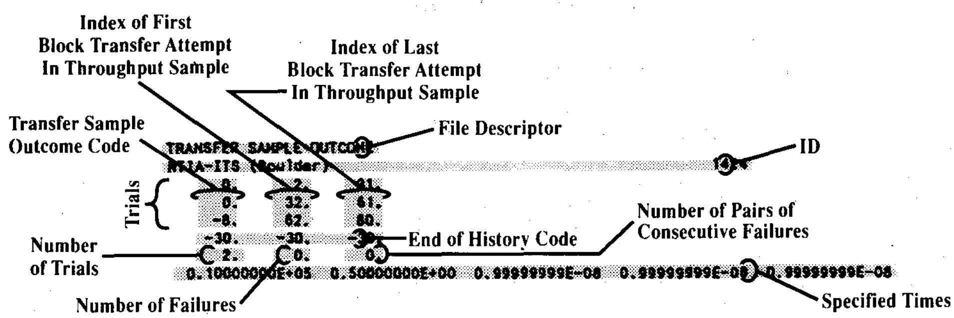
Figure 31. Examples of bit outcome (B10) and block outcome (B20) files.

- the batch identification,
- the Block Transfer Time and the user portion of Block Transfer Time for each successful trial,
- an unsuccessful block transfer attempt code for each unsuccessful trial,
- the end of history code (i.e., -30.),
- the number of block transfer attempts and both the number of failures and the number of pairs of consecutive failures for three types of failure outcomes: Incorrect Block outcome, Lost Block outcome, and Extra Block outcome, and
- the specified Block Transfer Time and the specified User Fraction of Block Transfer Time (both from the SPI file).

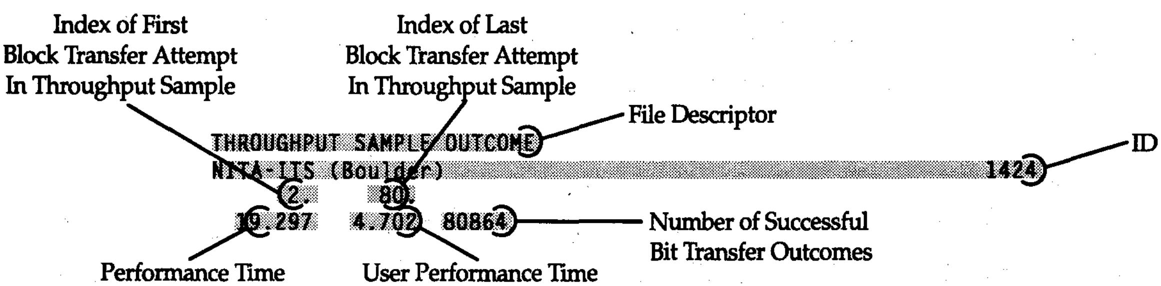
c. Transfer Availability Trial Outcome File

B30 is the file that contains transfer availability trial outcomes. It is called the transfer sample outcome file in Figure 32a because this is the name given to it in Volume 4. Each line contains

- the file descriptor,
- the batch identification,
- the transfer availability trial outcome code, the index of the first block transfer attempt in the transfer sample, and the index of the last block transfer attempt in the transfer sample,
- the end of history (i.e., -30.), and
- a failure summary containing the number of transfer availability trials, the number of Transfer Denial outcomes, and the number of pairs of consecutive Transfer Denial outcomes, and
- the specified values for the following five performance parameters: User Information Bit Transfer Rate, User Fraction of Input/Output Time, Bit Error Probability, Bit Loss Probability, and Extra Bit Probability.



a. Transfer Availability Outcome (B30) File.



b. Throughput Outcome (B40) File.

Figure 32. Examples of transfer availability outcome (B30) and throughput outcome (B40) files.

d. Throughput Trial Outcome File

B40 is the file that contains the throughput trial outcome. It is called the throughput sample outcome file in Figure 32b because this is the name given to it in Volume 4. Each line contains

- the file descriptor,
- the batch identification,
- the index of the first block transfer attempt and the index of the last block transfer attempt, and
- the Input/Output Time, the user portion of Input/Output Time, and the number of Successful Bit Transfer outcomes.

C. Calculate User Information Transfer Performance Parameters

The user information transfer function is divided (in this volume) into three functions: the bit transfer function, the block transfer function, and the transfer sample transfer function. The calculations of the performance parameters are discussed according to their function.

1. Bit Transfer Performance Parameters

Bit Error Probability, Extra Bit Probability, and Lost Bit Probability are calculated by the program **ANALYZ** using the outcomes in the **B10** file. They are calculated according to the definitions in Section 1.3.2. The optional Bit Misdelivery Probability is not calculated.

2. Block Transfer Performance Parameters

Block Transfer Time, User Fraction of Block Transfer Time, Block Error Probability, Extra Block Probability, and Lost Block Probability are calculated by the program **ANALYZ** and subroutine **ANCILL** using the outcomes in the **B20** file. They are calculated according to the definitions in Section 1.3.2. The optional Block Misdelivery Probability is not calculated.

3. Transfer Sample Transfer Performance Parameters

a. User Information Bit Transfer Rate

This performance parameter is calculated both for a single transfer sample to calculate one of the supported performance parameters and for a series of transfer samples to calculate a measure of throughput.

(i.) Single Transfer Sample

For a single transfer sample having the specified minimum number of bit transfer attempts, the Input/Output Time is calculated, and the number of Successful Bit Transfers is counted. This version of the User Information Bit Transfer Rate is calculated to serve as a supported performance parameter. It is calculated by the program **ANALYZ** using the outcomes in the **B30** file. This implementation conforms to both versions of ANS X3.102: it meets the requirement of the original version, and the revised version does not require that the transfer sample have either a specified minimum Input/Output Time or a specified minimum number of bit transfer attempts.

(ii.) Series of Transfer Samples

For the longest transfer sample having the specified minimum number of bits, the Input/Output Time is calculated, and the number of Successful Bit Transfer attempts is counted.[54] This version of the User Information Bit Transfer Rate is calculated by program **ANALYZ** using the outcomes in the **B40** file. The original version of ANS X3.102 does not define a "long-term" version of User Information Bit Transfer Rate, but it alludes to the advisability of calculating one. The revised version of ANS X3.102 specifies a minimum Input/Output Time for each transfer sample in a series (a condition that renders them transfer availability trials), and it specifies that the number of Successful Bit Transfer outcomes from transfer availability trials having Successful Transfer outcomes be counted. (The NTIA implementation does not require the transfer sample to have the Successful Transfer outcome.)

---

[54]As noted earlier, a transfer sample can contain a series of transfer samples as well as nested transfer samples.

Hence the NTIA implementation exceeds the requirements of the original version of ANS X3.102 (which do not exist), but does not meet the requirements of the revised version of ANS X3.102.

b. User Fraction of Input/Output Time

The User Fraction of Input/Output Time is the ancillary performance parameter for the User Information Bit Transfer Rate.

(i.) Single Transfer Sample

This ancillary performance parameter is calculated from the times of the single transfer sample having the specified minimum number of bits. This performance parameter is calculated by subroutine **UIOTIM**.

(ii.) Series of Transfer Samples

This ancillary performance parameter is calculated from the times of the longest transfer sample having the specified minimum number of bits. This performance parameter is calculated by subroutine **UIOTIM**.

c. Transfer Denial Probability

Transfer Denial Probability is calculated by the program **ANALYZ** using the outcomes in the **B30** file. It is calculated according to the definition in Section 1.3.2, but the Transfer Denial outcomes are determined from transfer availability trials that are determined from the specified minimum number of bits (original ANS X3.102) rather than from the specified minimum Input/Output Time (revised ANS X3.102). The supported performance parameters used to determine the outcomes of transfer availability trials are determined according to either/both versions of ANS X3.102. Specifically, the User Information Bit Transfer Rate, when used as a supported performance parameter, is calculated from a single transfer sample.

5.4.3 Assess Disengagement Performance

Subroutine **DISENG** assesses disengagement performance. It identifies disengagement attempts, determines disengagement outcomes, and calculates the disengagement performance parameters.

A. Identify Disengagement Attempts

Disengagement starts with the reference event, Disengagement Request, and normally ends with the reference event, Disengagement Confirmation.

B. Determine Disengagement Outcomes

1. Disengagement Outcome Scheme

Figure 33 is a flowchart showing how disengagement outcomes are determined. It includes a nonclassified outcome that is not specified by ANS X3.141 (i.e., Figure 16).

2. Disengagement Outcome Files

The above procedure produces the source disengagement outcome file (**D1O**) and the destination disengagement outcome file (**D2O**) which are shown in Figures 34a and 34b, respectively.

a. Source Disengagement Outcome File

Each line of D1O contains

- the file descriptor,
- the batch identification,
- the Source Disengagement Time and the user portion of Source Disengagement Time for each successful trial,
- an unsuccessful source disengagement attempt code for each unsuccessful trial,
- the end of history code (i.e., -30.),
- the number of source disengagement attempts, the number of failures and the number of pairs of consecutive failures for one type of failure: Source Disengagement Denial Probability, and

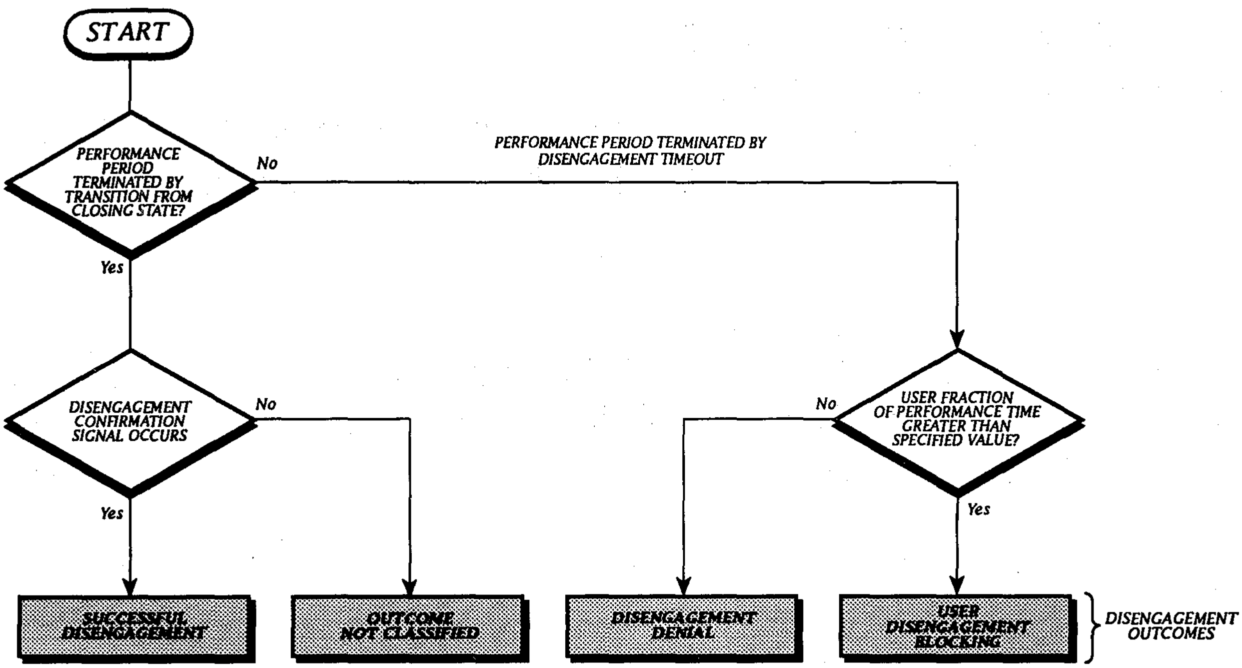
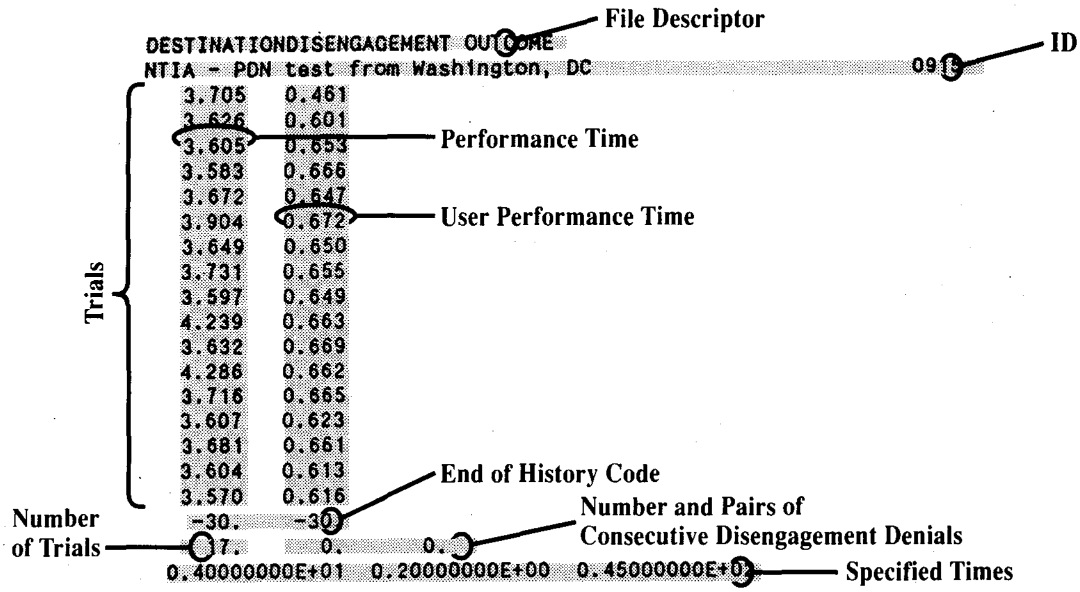
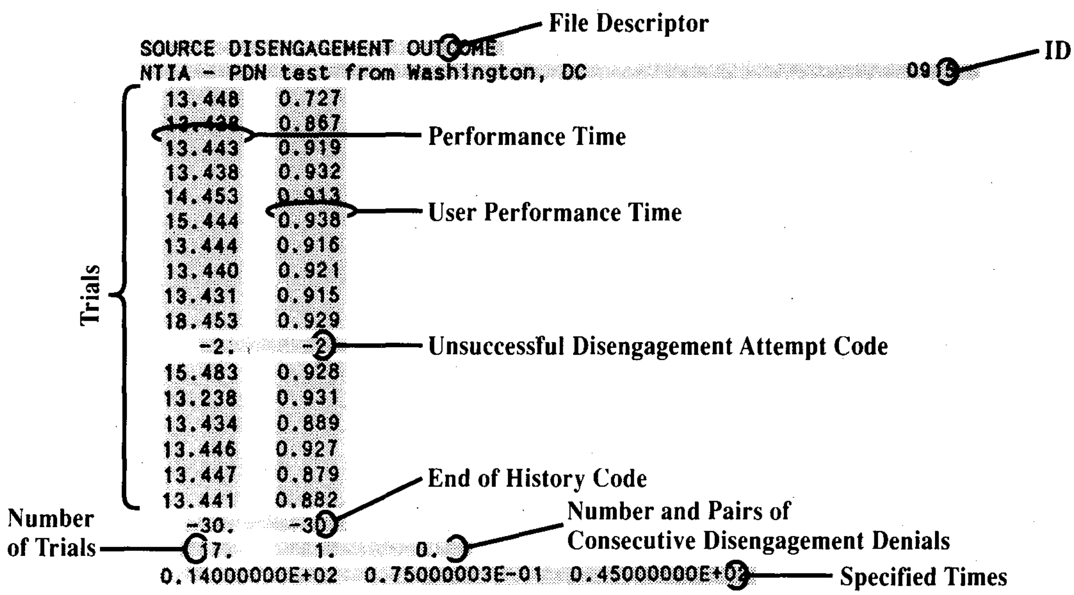


Figure 33. Flowchart that determines disengagement outcomes according to NTIA.



a. D10 File.



b. D20 File

Figure 34. Examples of source disengagement outcome (D10) and destination disengagement outcome (D20) files.

- the specified Source Disengagement Time, the specified user fraction of Source Disengagement Time, and the specified Access Time (all from the **SPI** file).

b. Destination Disengagement Outcome File

Each line of **D20** contains

- the file descriptor,
- the batch identification,
- the Destination Disengagement Time and the user portion of Destination Disengagement Time for each successful trial,
- an unsuccessful destination disengagement attempt code for each unsuccessful trial,
- the end of history code (i.e., -30.),
- the number of destination disengagement attempts, the number of failures and the number of pairs of consecutive failures for one type of failure: Destination Disengagement Denial Probability, and
- the specified Destination Disengagement Time, the specified user fraction of Destination Disengagement Time, and the specified Access Time (all from the **SPI** file).

C. Calculate Disengagement Performance Parameters

The three performance parameters from each end user from the disengagement function are calculated by program **ANALYZ** and subroutine **ANCILL** using the outcomes in the **D10** and **D20** files. They are calculated as defined in Section 1.3.3.

The seven outcome files (i.e., **AC0**, **B10**, **B20**, **B30**, **B40**, **D10** and, **D20**) are required by the data analysis phase of the NTIA design.

5.4.4 Allocate Performance Time

Performance periods are divided into a sequence of intervals by communication state transitions.

Subroutine **ANCILL** evaluates user performance time for the access, block transfer, and disengagement functions. The relevant interfaces are the same for all communication state intervals during the performance period of these functions.

Subroutine **UITIM** evaluates transfer availability trials and throughput trials. For performance periods of Input Time and Output Time that are associated with transfer availability trials and throughput trials, the relevant interfaces differ from one interval to another.

These two subroutines add the portions of user responsible times during a data communication function. The responsibility state for an interval is determined by the local responsibility states at the source and destination interfaces as specified in ANS X3.141. See Section 2.4.4.

5.5 Produce Performance Assessment Summary

Program **EPILOG** creates an assessment summary file (called **SUM**). In user information transfer tests, it compares source and destination user data and analyzes them to determine bit transfer and block transfer outcomes.

5.5.1 Access Function

Figure 35 is an example of the access assessment summary file. It provides performance statistics, estimated performance parameter values, and reduction specifications for each test (i.e., set of trials). The data are the same as provided in the **ACO** file except the performance parameter values have been estimated.

5.5.2 User Information Function

Figure 36 is an example of the three-part user information transfer assessment summary file. It provides the correlation summary and the correlator specifications (from the **SPI** file). Then it provides performance statistics (provided in files **B1O**, **B2O**, **B3O**, and **B4O**), estimated performance parameter values, and reduction specifications for each test (i.e., set of trials). The performance parameters have been estimated.

```
---------------------------------------------------------------
NTIA - PDN test from Washington, DC                        0915
---------------------------------------------------------------
```

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                            *
*                         ACCESS ASSESSMENT SUMMARY                          *
*                                                                            *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                            *
*                            -------------------                             *
*                            PERFORMANCE SUMMARY                             *
*                            -------------------                             *
*                                                                            *
*  ACCESS ATTEMPTS . . . . . . . . . . . . . . . . . . . . . .     20 (+)    *
*  'SUCCESSFUL ACCESS' OUTCOMES  . . . . . . . . . . . . . . .     17        *
*  'INCORRECT ACCESS' OUTCOMES . . . . . . . . . . . . . . . .      0        *
*  PAIRS OF SUCCESSIVE 'INCORRECT ACCESS' OUTCOMES . . . . . .      0        *
*  'ACCESS DENIAL' OUTCOMES  . . . . . . . . . . . . . . . . .      3        *
*  PAIRS OF SUCCESSIVE 'ACCESS DENIAL' OUTCOMES  . . . . . . .      0        *
*  'ACCESS OUTAGE' OUTCOMES  . . . . . . . . . . . . . . . . .      0        *
*  PAIRS OF SUCCESSIVE 'ACCESS OUTAGE' OUTCOMES  . . . . . . .      0        *
*                                                                            *
*  (+) THIS NUMBER EXCLUDES ATTEMPTS THAT FAIL DUE TO USER NONPERFORMANCE    *
*                                                                            *
*                                                                            *
*                   ------------------------------------                     *
*                   MEASURED PERFORMANCE PARAMETER VALUES                    *
*                   ------------------------------------                     *
*                                                                            *
*  ACCESS TIME . . . . . . . . . . . . . . . . . . . . . .     45.477 SEC    *
*  USER FRACTION OF ACCESS TIME  . . . . . . . . . . . . .     0.0362        *
*                                                                            *
*  INCORRECT ACCESS PROBABILITY  . . . . . . . . . . . . .          0        *
*  ACCESS DENIAL PROBABILITY . . . . . . . . . . . . . . 1.5 X 10(-01)       *
*  ACCESS OUTAGE PROBABILITY . . . . . . . . . . . . . . .          0        *
*                                                                            *
*                                                                            *
*                         -----------------------                            *
*                         REDUCTION SPECIFICATIONS                           *
*                         -----------------------                            *
*                                                                            *
*  SPECIFIED ACCESS TIME . . . . . . . . . . . . . . . . .     45.000 SEC    *
*  SPECIFIED USER FRACTION OF ACCESS TIME  . . . . . . . .     0.0500        *
*                                                                            *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 35. Example of access assessment summary file.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*               USER INFORMATION TRANSFER  ASSESSMENT SUMMARY                 *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*                          -------------------                                *
*                          CORRELATION SUMMARY                                *
*                          -------------------                                *
*                                                                             *
*  CORRELATOR OUTPUT BLOCKS . . . . . . . . . . . . . . . . .        80       *
*                                                                             *
*  BIT COMPARISON OUTCOMES (BCOS) . . . . . . . . . . . . . . .   81944       *
*  'CORRECT' BCOS . . . . . . . . . . . . . . . . . . . . . . .   81888       *
*  'INCORRECT' BCOS . . . . . . . . . . . . . . . . . . . . . .       8       *
*  'UNDELIVERED' BCOS . . . . . . . . . . . . . . . . . . . . .      24       *
*  'EXTRA' BCOS . . . . . . . . . . . . . . . . . . . . . . . .      24       *
*                                                                             *
*  SOURCE USER INFORMATION BLOCKS TRANSMITTED . . . . . . . . .      80       *
*  SOURCE USER INFORMATION BLOCKS CORRELATED . . . . . . . . . .     80       *
*                                                                             *
*  SOURCE USER INFORMATION BITS TRANSMITTED . . . . . . . . . .   81920       *
*  SOURCE USER INFORMATION BITS CORRELATED . . . . . . . . . . .  81920       *
*                                                                             *
*  DESTINATION USER INFORMATION BLOCKS RECEIVED . . . . . . . .      80       *
*  DESTINATION USER INFORMATION BLOCKS CORRELATED . . . . . . .      80       *
*                                                                             *
*  DESTINATION USER INFORMATION BITS RECEIVED . . . . . . . . .   81920       *
*  DESTINATION USER INFORMATION BITS CORRELATED . . . . . . . .   81920       *
*                                                                             *
*                                                                             *
*                       -------------------------                             *
*                       CORRELATOR SPECIFICATIONS                             *
*                       -------------------------                             *
*                                                                             *
*  USER INFORMATION WINDOW SIZE . . . . . . . . . . . . . . . .      16 BITS  *
*                                                                             *
*  MAXIMUM DATA SHIFT EXECUTED IN BIT ERROR                                   *
*    IDENTIFICATION ALGORITHM . . . . . . . . . . . . . . . . .     256 BITS  *
*  MAXIMUM DATA SHIFT EXECUTED IN UNDELIVERED BIT                             *
*    IDENTIFICATION ALGORITHM . . . . . . . . . . . . . . . . .    8192 BITS  *
*  MAXIMUM DATA SHIFT EXECUTED IN EXTRA BIT                                   *
*    IDENTIFICATION ALGORITHM . . . . . . . . . . . . . . . . .    8192 BITS  *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 36. (Part 1). **Example of user information transfer assessment summary file.**

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*           USER INFORMATION TRANSFER  ASSESSMENT SUMMARY                     *
*                            (CONTINUED)                                      *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*                       ----------------------                                *
*                       PERFORMANCE  STATISTICS                               *
*                       ----------------------                                *
*                                                                             *
*  BIT TRANSFER ATTEMPTS . . . . . . . . . . . . . . . . . .   81944 (+)     *
*  'SUCCESSFUL BIT TRANSFER' OUTCOMES  . . . . . . . . . . .   81888         *
*  'INCORRECT BIT' OUTCOMES  . . . . . . . . . . . . . . . .       8         *
*  PAIRS OF SUCCESSIVE 'INCORRECT BIT' OUTCOMES  . . . . . .       2         *
*  'LOST BIT' OUTCOMES . . . . . . . . . . . . . . . . . . .      24         *
*  PAIRS OF SUCCESSIVE 'LOST BIT' OUTCOMES . . . . . . . . .      23         *
*  'EXTRA BIT' OUTCOMES  . . . . . . . . . . . . . . . . . .      24         *
*  PAIRS OF SUCCESSIVE 'EXTRA BIT' OUTCOMES  . . . . . . . .      23         *
*                                                                             *
*  BLOCK TRANSFER ATTEMPTS . . . . . . . . . . . . . . . . .      80 (+)     *
*  'SUCCESSFUL BLOCK TRANSFER' OUTCOMES  . . . . . . . . . .      78         *
*  'INCORRECT BLOCK' OUTCOMES  . . . . . . . . . . . . . . .       2         *
*  PAIRS OF SUCCESSIVE 'INCORRECT BLOCK' OUTCOMES  . . . . .       0         *
*  'LOST BLOCK' OUTCOMES . . . . . . . . . . . . . . . . . .       0         *
*  PAIRS OF SUCCESSIVE 'LOST BLOCK' OUTCOMES . . . . . . . .       0         *
*  'EXTRA BLOCK' OUTCOMES  . . . . . . . . . . . . . . . . .       0         *
*  PAIRS OF SUCCESSIVE 'EXTRA BLOCK' OUTCOMES  . . . . . . .       0         *
*                                                                             *
*  TRANSFER SAMPLES  . . . . . . . . . . . . . . . . . . . .       2 (+)     *
*  'TRANSFER DENIAL' OUTCOMES  . . . . . . . . . . . . . . .       0         *
*  PAIRS OF SUCCESSIVE 'TRANSFER DENIAL' OUTCOMES  . . . . .       0         *
*                                                                             *
*  INPUT TIME FOR THROUGHPUT SAMPLE  . . . . . . . . . . . .  17.769 SEC     *
*  USER INPUT TIME FOR THROUGHPUT SAMPLE . . . . . . . . . .   4.702 SEC     *
*  OUTPUT TIME FOR THROUGHPUT SAMPLE . . . . . . . . . . . .  19.297 SEC     *
*  USER OUTPUT TIME FOR THROUGHPUT SAMPLE  . . . . . . . . .   2.054 SEC     *
*  BIT TRANSFER ATTEMPTS IN THROUGHPUT SAMPLE  . . . . . . .   80920         *
*  'SUCCESSFUL BIT TRANSFER' OUTCOMES IN THROUGHPUT SAMPLE .   80864         *
*  BLOCK TRANSFER ATTEMPTS IN THROUGHPUT SAMPLE  . . . . . .      79         *
*                                                                             *
*  (+) THIS NUMBER EXCLUDES ATTEMPTS THAT FAIL DUE TO USER NONPERFORMANCE     *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 36. (Part 2). **Example of user information transfer assessment summary file.**

```
-----------------------------------------------------------------
NTIA-ITS (Boulder)                                           1424
-----------------------------------------------------------------

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                              *
*          USER INFORMATION TRANSFER  ASSESSMENT SUMMARY                       *
*                            (CONTINUED)                                       *
*                                                                              *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                              *
*                 -------------------------------------                        *
*                 MEASURED PERFORMANCE PARAMETER VALUES                        *
*                 -------------------------------------                        *
*                                                                              *
*  BIT ERROR PROBABILITY . . . . . . . . . . . . . . . . 9.8 X 10(-05)         *
*  BIT LOSS PROBABILITY  . . . . . . . . . . . . . . . . 2.9 X 10(-04)         *
*  EXTRA BIT PROBABILITY . . . . . . . . . . . . . . . . 2.9 X 10(-04)         *
*                                                                              *
*  BLOCK TRANSFER TIME . . . . . . . . . . . . . . . .          2.461 SEC      *
*  USER FRACTION OF BLOCK TRANSFER TIME  . . . . . . .         0.1001          *
*                                                                              *
*  BLOCK ERROR PROBABILITY . . . . . . . . . . . . . . . 2.5 X 10(-02)         *
*  BLOCK LOSS PROBABILITY  . . . . . . . . . . . . . .               0         *
*  EXTRA BLOCK PROBABILITY . . . . . . . . . . . . . .               0         *
*                                                                              *
*  TRANSFER DENIAL PROBABILITY . . . . . . . . . . . .               0         *
*                                                                              *
*  USER INFORMATION BIT TRANSFER RATE  . . . . . . . .          4190 BPS       *
*  USER FRACTION OF INPUT/OUTPUT TIME  . . . . . . . .         0.2437          *
*                                                                              *
*                                                                              *
*                       ------------------------                               *
*                       REDUCTION  SPECIFICATIONS                              *
*                       ------------------------                               *
*                                                                              *
*  SPECIFIED BLOCK TRANSFER TIME . . . . . . . . . . .          3.000 SEC      *
*  SPECIFIED USER FRACTION OF BLOCK TRANSFER TIME  . .         0.5000          *
*                                                                              *
*  SPECIFIED USER INFORMATION BIT TRANSFER RATE                                *
*    FOR TRANSFER SAMPLE . . . . . . . . . . . . . . .          10000 BPS      *
*  SPECIFIED USER FRACTION OF INPUT/OUTPUT TIME                                *
*    FOR TRANSFER SAMPLE . . . . . . . . . . . . . . .         0.5000          *
*                                                                              *
*  SPECIFIED BIT ERROR PROBABILITY FOR TRANSFER SAMPLE . 1.0 X 10(-08)         *
*  SPECIFIED BIT LOSS PROBABILITY FOR TRANSFER SAMPLE  . 1.0 X 10(-08)         *
*  SPECIFIED EXTRA BIT PROBABILITY FOR TRANSFER SAMPLE . 1.0 X 10(-08)         *
*                                                                              *
*  MINIMUM NUMBER OF BIT TRANSFER ATTEMPTS                                     *
*    IN TRANSFER SAMPLE  . . . . . . . . . . . . . . .          30000          *
*                                                                              *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 36. (Part 3). **Example of user information transfer assessment summary file.**

5.5.3 Disengagement Function

Figures 37 and 38 are examples of the source disengagement and destination disengagement assessment summary files, respectively. They provide performance statistics, estimated performance parameter values, and reduction specifications for each test (i.e., set of trials).

```
-----------------------------------------------------------------
NTIA - PDN test from Washington, DC                          0915
-----------------------------------------------------------------

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*                                 SOURCE                                      *
*                    DISENGAGEMENT  ASSESSMENT SUMMARY                        *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*                         ----------------------                              *
*                         PERFORMANCE STATISTICS                              *
*                         ----------------------                              *
*                                                                             *
*  DISENGAGEMENT ATTEMPTS . . . . . . . . . . . . . . . . . . .    17 (+)     *
*  'SUCCESSFUL DISENGAGEMENT' OUTCOMES . . . . . . . . . . . . .    16        *
*  'DISENGAGEMENT DENIAL' OUTCOMES . . . . . . . . . . . . . . .     1        *
*  PAIRS OF SUCCESSIVE 'DISENGAGEMENT DENIAL' OUTCOMES . . . .       0        *
*                                                                             *
*  (+) THIS NUMBER EXCLUDES ATTEMPTS THAT FAIL DUE TO USER NONPERFORMANCE     *
*                                                                             *
*                                                                             *
*                  ------------------------------------                       *
*                  MEASURED PERFORMANCE PARAMETER VALUES                      *
*                  ------------------------------------                       *
*                                                                             *
*  DISENGAGEMENT TIME . . . . . . . . . . . . . . . . .         14.058 SEC    *
*  USER FRACTION OF DISENGAGEMENT TIME . . . . . . . . .        0.0641        *
*                                                                             *
*  DISENGAGEMENT DENIAL PROBABILITY . . . . . . . . .  5.9 X 10(-02)          *
*                                                                             *
*                                                                             *
*                        ------------------------                             *
*                        REDUCTION SPECIFICATIONS                             *
*                        ------------------------                             *
*                                                                             *
*  SPECIFIED DISENGAGEMENT TIME . . . . . . . . . . .          14.000 SEC     *
*  SPECIFIED USER FRACTION OF DISENGAGEMENT TIME . . .         0.0750         *
*                                                                             *
*  SPECIFIED ACCESS TIME . . . . . . . . . . . . . . .         45.000 SEC     *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 37. Example of source disengagement assessment summary file.

```
-----------------------------------------------------------------
NTIA - PDN test from Washington, DC                          0915
-----------------------------------------------------------------


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                           *
*                               DESTINATION                                 *
*                    DISENGAGEMENT  ASSESSMENT SUMMARY                      *
*                                                                           *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                           *
*                          ----------------------                           *
*                          PERFORMANCE STATISTICS                           *
*                          ----------------------                           *
*                                                                           *
*  DISENGAGEMENT ATTEMPTS . . . . . . . . . . . . . . . . . . .    17 (+)   *
*  'SUCCESSFUL DISENGAGEMENT' OUTCOMES . . . . . . . . . . . . .   17       *
*  'DISENGAGEMENT DENIAL' OUTCOMES . . . . . . . . . . . . . . .    0       *
*  PAIRS OF SUCCESSIVE 'DISENGAGEMENT DENIAL' OUTCOMES . . . .      0       *
*                                                                           *
*  (+) THIS NUMBER EXCLUDES ATTEMPTS THAT FAIL DUE TO USER NONPERFORMANCE   *
*                                                                           *
*                                                                           *
*                   -------------------------------------                   *
*                   MEASURED PERFORMANCE PARAMETER VALUES                   *
*                   -------------------------------------                   *
*                                                                           *
*  DISENGAGEMENT TIME . . . . . . . . . . . . . . . . .         3.730 SEC   *
*  USER FRACTION OF DISENGAGEMENT TIME . . . . . . . .         0.1707       *
*                                                                           *
*  DISENGAGEMENT DENIAL PROBABILITY . . . . . . . . .              0       *
*                                                                           *
*                                                                           *
*                         ------------------------                          *
*                         REDUCTION SPECIFICATIONS                          *
*                         ------------------------                          *
*                                                                           *
*  SPECIFIED DISENGAGEMENT TIME . . . . . . . . . . . .         4.000 SEC   *
*  SPECIFIED USER FRACTION OF DISENGAGEMENT TIME . . .         0.2000       *
*                                                                           *
*  SPECIFIED ACCESS TIME . . . . . . . . . . . . . . .         45.000 SEC   *
*                                                                           *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 38. Example of destination disengagement assessment summary file.

6. ANALYZE THE DATA

A test consists of a number of trials, and an experiment usually consists of a number of tests. The NTIA design of the analysis phase allows the performance parameters to be analyzed in one of four ways, depending upon the objective of the experiment:[55] estimation, acceptance tests, comparison tests, and tests to determine whether the levels of a variable condition affect a performance parameter (i.e., whether the variable condition is a factor).

The NTIA design considers dependence between trials by assuming that dependence occurs between consecutive trials only (i.e., a first-order Markov chain). Considering dependence does not affect the bias of an estimate, but it does improve the estimate of its precision. Most dependence results in positive autocorrelation which widens the confidence interval. Hence, when dependence exists, more trials are usually required to achieve a specified precision (i.e., width).

For the purpose of experiment design and analysis, the performance parameters are considered to be one of the following types of random variables: delays, rates, or failures (re: Table 4b). Depending upon the objective of the experiment, either single tests or multiple tests should be analyzed.

The theory for analysis of data communication systems is discussed in a book authored by M. J. Miles and E. L. Crow (to be published).

6.1 Introduction

The seven outcome files from data reduction (i.e., **ACO**, **D1O**, **D2O**, **B1O**, **B2O**, **B3O**, and **B4O**) are required by the data analysis phase. They are used for two purposes in data analysis:

- Analysis of a Single Test. The **do** shell script can be activated to use these files to analyze each test.
- Analysis of Multiple Tests. UNIX[tm] utilities edit outcome files from single tests for multiple test analysis:

[55]In all cases, it is performance parameters that are analyzed - not functions of performance parameters nor combinations of performance parameters. Analysis could be extended to include functions and combinations, but such extensions would be subjective and beyond ANS X3.102. See Appendix C of ANS X3.102.

- Delays. For delay performance parameters, these files are **nnnnac**, **nnnnd1**, **nnnnd2**, and **nnnnb2** where **nnnn** is the test number.
- Rates. For rate performance parameters, this file is **thrput**.
- Failures. For failure probability performance parameters, these files are **failac**, **faild1**, **faild2**, **failb1**, **failb2**, **failb3**, and **failb4**.

6.1.1 Analysis of A Single Test

Analysis of single tests can be implemented either by a shell script or by an operator.

A. Shell Script Implementation

Shell script implementation of analysis of single tests is initiated during the data extraction phase by typing

do nnnn

where **nnnn** is the test number: Extracted data that have been consolidated in one computer are converted from binary to ASCII code, reduced, and analyzed. Figure 39 is a sample of the analysis of failure probabilities from an access-disengagement test when the **do** shell script is activated.

B. Operator Implementation

Reduced data can be analyzed by typing

star

and then typing the responses and data as requested by this interactive FORTRAN program: Specifically,

- delay data can be entered manually or by a file,[56]
- failure probability data must be entered manually, and

[56]Data entered manually can come from any source. If the reduced data from an NTIA implemented test are to be analyzed by an operator, they can be found in a file in directory **/usr/data/5a**; files containing data from another source should be placed in directory **/usr/net/src/d5** (which also contains **star**).

MEASUREMENT RESULTS SUMMARY

NTIA-ITS (Boulder) 2218

PERFORMANCE PARAMETER	SAMPLE SIZE	ESTIMATED VALUE	CONFIDENCE LEVEL (PERCENT)	LOWER CONFIDENCE LIMIT	UPPER CONFIDENCE LIMIT
ACCESS TIME	33	.47858E+02	90	.47333E+02	.48383E+02
			95	.47227E+02	.48490E+02
USER FRACTION OF ACCESS TIME	33	.26687E-02	90	.18527E-02	.34847E-02
			95	.16964E-02	.36409E-02
DISENGAGEMENT TIME (SOURCE)	28	.89793E+01	90	.87887E+01	.91698E+01
			95	.87498E+01	.92087E+01
USER FRACTION OF DISENGAGEMENT TIME (SOURCE)	28	.87136E-02	90	.85151E-02	.89122E-02
			95	.84770E-02	.89503E-02
DISENGAGEMENT TIME (DESTINATION)	33	.86597E+00	90	.81650E+00	.91544E+00
			95	.80648E+00	.92546E+00
USER FRACTION OF DISENGAGEMENT TIME (DESTINATION)	33	.70513E-01	90	.66392E-01	.74634E-01
			95	.65602E-01	.75423E-01

ESTIMATED PERFORMANCE TIMES ARE EXPRESSED IN SECONDS

Figure 39. Example from shell script implementation of analysis of failure probability parameters from an access disengagement test.

- rate data can be entered manually or by a file. (However, if the NTIA implementation is used, data from a single test cannot be analyzed because there is at most one trial per test.)

Figure 40 is a structured design diagram of the operator implementation of **star** to analyze a test. The output is the sample size, the estimate of the performance parameter, its 90% or 95% confidence limits, its standard deviation, and its autocorrelation of lag 1 (i.e., a measure of dependence existing between consecutive trials).

6.1.2 Analysis of Multiple Tests

Each test is conducted for a certain combination of levels of variable conditions. The experimenter can select a group of tests and pool their data. The rationale for selecting the tests depends upon the objective. However, the standard deviations of the primary performance parameter from each test should be rather similar. The following four subsections discuss pooling, an example of analysis of multiple tests, implementation by a shell script, and implementation by an operator.

A. Pooling

Program **star** pools data for a primary performance parameter in three ways: The trials, the test means, and the means of the levels of the variable condition.

Program **star** determines the acceptability of pooling trials and test means:

- <u>It Determines if Trials Come from the Same Population</u>. If the null hypothesis of equal means of tests may be accepted, trials from all tests are considered to come from the same population and may be pooled.

- <u>It Determines if Test Means Come from the Same Population</u>. If the null hypothesis of equal means of levels may be accepted, means from all tests are considered to come from the same population and may be pooled.

There is no test to determine if the means of the levels come from the same population. Pooling trials results in the shortest confidence interval (i.e.,

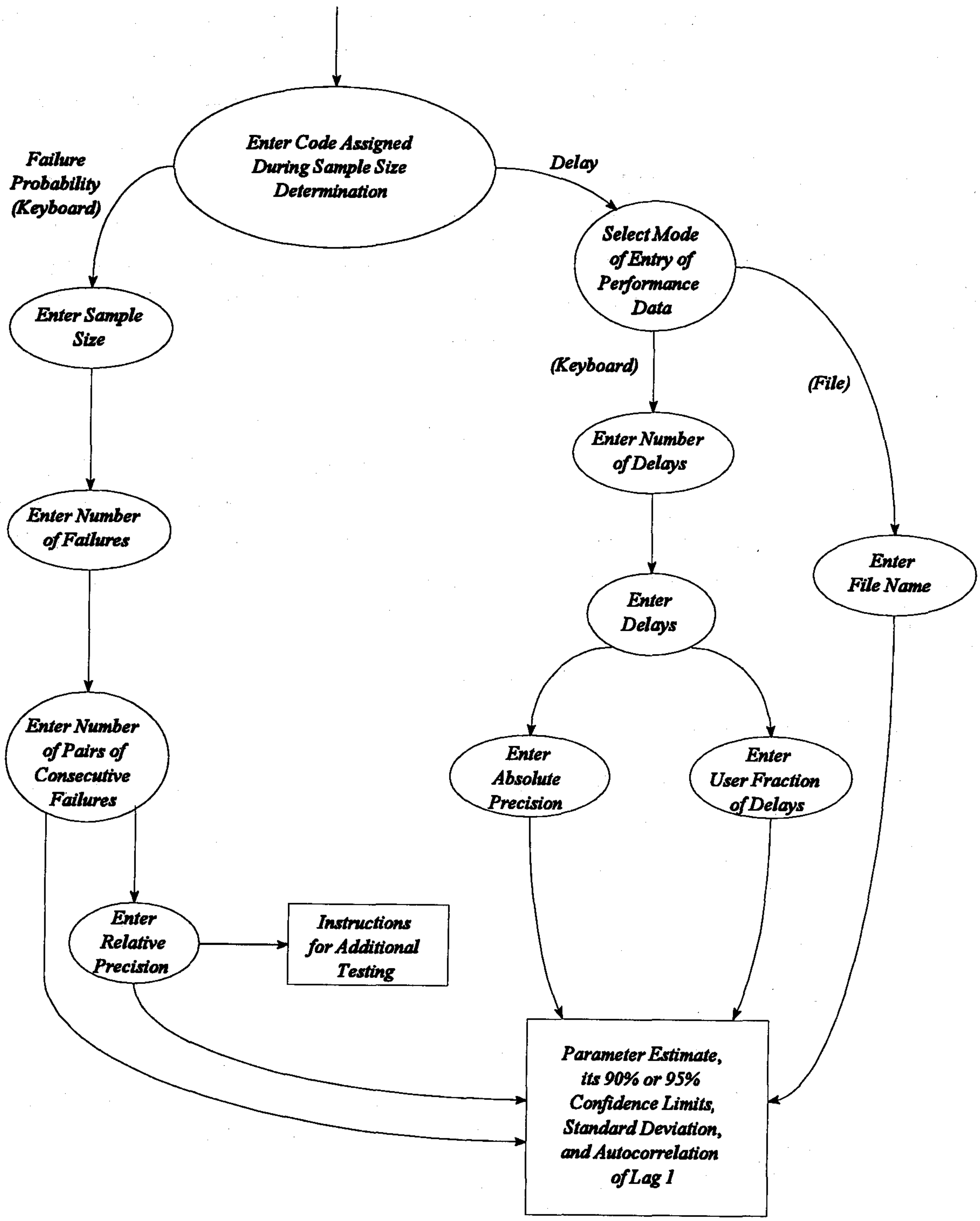


Figure 40. Structured design diagram of operator implementation of analysis of single tests.

the most precision), pooling test means results in the next shorter confidence interval, and pooling level means results in the longest confidence interval. Figure 41 is a flowchart illustrating the hypothesis testing procedure.

B. Example of Data Analysis

Figure 42 shows the analysis of 11 tests of Access Time and User Fraction of Access Time. The upper portion of this figure lists data from each of the single tests, and the lower portion lists data from the analysis of multiple tests.

1. Single Test Data

The data from each of the single tests consist of identification of the tests, their sample sizes, and estimates of their means and standard deviations for both the primary and its ancillary performance parameter. Identification includes the test number and the level of six variable conditions. The six variable conditions are

- Source Site (Fort Worth, Seattle, and Washington DC),
- Network (A),
- Day of Week (Monday, Tuesday, Wednesday, Thursday, and Friday),
- Time of Day (periods 1 through 6 - each consisting of 4 hours),
- Interaccess Delay (55 s), and
- Destination Site (Boulder).

In this group of tests, Network, Interaccess Delay, and Destination Site have only one level. The variable condition, Source Site, is selected for pooling. That is, it is to be determined if Source Site is a factor (at least for these three qualitative levels).

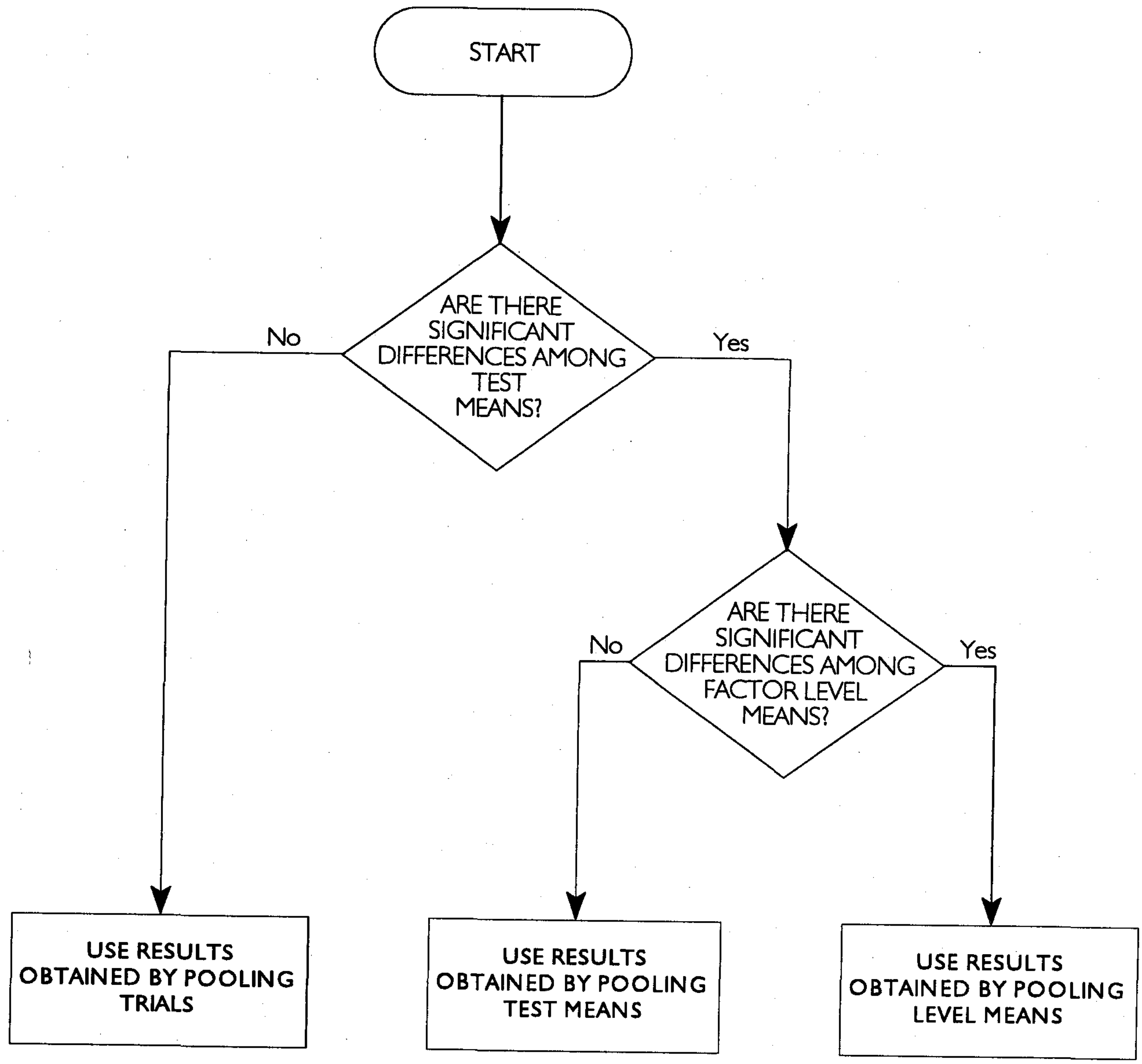


Figure 41. Flowchart that determines pooling for analysis of multiple tests.

Analysis of Multiple Tests

Access Time
Variable Condition 1

Thu Jan 26 14:06:14 MST 1989

Test	--- Variable Conditions ---	Trials	---- Times ---- Mean	Std Dev	User Fractions Mean	Std Dev
775	ftw netA fri 1 A55 bol	20	38.291	1.608	0.0397	0.0199
823	sea netA fri 2 A55 bol	20	42.439	1.527	0.0339	0.0047
815	sea netA fri 6 A55 bol	20	41.576	1.269	0.0352	0.0044
835	sea netA mon 3 A55 bol	15	42.954	1.325	0.0345	0.0053
858	sea netA thu 1 A55 bol	20	42.284	1.338	0.0345	0.0053
876	sea netA thu 4 A55 bol	20	42.313	2.197	0.0350	0.0064
811	sea netA thu 5 A55 bol	19	41.163	1.015	0.0373	0.0065
997	wdc netA thu 3 A55 bol	17	41.751	2.198	0.0356	0.0075
928	wdc netA tue 1 A55 bol	20	44.500	4.380	0.0332	0.0068
952	wdc netA tue 5 A55 bol	20	39.813	1.625	0.0368	0.0043
978	wdc netA wed 4 A55 bol	18	42.304	1.820	0.0351	0.0054

TIMES (W) AND FRACTION OF TIMES (V)

NUMBER OF TRIALS = 209
NUMBER OF TESTS = 11
NUMBER OF LEVELS = 3

WEIGHTED AVERAGE AUTOCORRELATION COEFFICIENT
OF LAG 1 OVER THE 11 TESTS = .3927E+00 #
AVERAGE AUTOCORRELATION COEFFICIENT
OF LAG 1 OVER THE 209 TRIALS = .4998E+00 @

	EFFECTIVE DEGREES OF FREEDOM		F STAT.	F DIST. (5%)		95% LOWER CONFIDENCE LIMIT	ESTIMATE OF THE MEAN	95% UPPER CONFIDENCE LIMIT	
AMONG TRIALS	80	10	.4961E+01	.1963E+01	W	.4112E+02	.4173E+02	.4234E+02	
	-	-	-	-	V	.3423E-01	.3547E-01	.3671E-01	
AMONG TESTS	8	2	.4011E+01	.4460E+01	W	.4067E+02	.4176E+02	.4286E+02	*
	-	-	-	-	V	.3416E-01	.3551E-01	.3686E-01	
AMONG LEVELS	-	-	-	-	W	.3536E+02	.4083E+02	.4631E+02	
	-	-	-	-	V	.2953E-01	.3676E-01	.4417E-01	

USED TO DETERMINE THE EFFECTIVE DEGREES OF FREEDOM FOR THE F TEST.
@ USED TO DETERMINE THE EFFECTIVE DEGREES OF FREEDOM FOR THE CONFIDENCE LIMITS.
* AT RIGHT OF UPPER CONFIDENCE LIMIT INDICTES THIS POOLING
IS ACCEPTABLE AT THE 5% LEVEL.

Figure 42. **Example of analysis of Access Time and User Fraction of Access time from multiple tests.**

2. Multiple Test Data

Data for analysis of multiple tests consist of three parts. Data follow the label,

TIMES (W) AND FRACTION OF TIMES (V)

a. Sizes from which to Determine the Degrees of Freedom

The first part of the output for multiple tests lists

- the number of trials,
- the number of tests, and
- the number of levels (for the selected variable condition, Source Site).

b. Autocorrelations of Lag 1 that Modify the Degrees of Freedom

The second part of the output for multiple tests lists

- the weighted average autocorrelation coefficient of lag 1 over the tests, and
- the average autocorrelation coefficient of lag 1 over the trials.

c. Multiple Test Analysis

The third part of the output for multiple tests lists the results of the three poolings of the primary performance parameter, Access Time. Access Time is indicated by **W** and User Fraction of Access Time is indicated by **V**. Each pooling lists the effective degrees of freedom,[57] the value of the F statistic, the value of the F distribution at the 5% point, the estimate of the lower 95% confidence limit, the estimate of the mean, and the estimate of the upper 95%

[57]The two degrees of freedom for each pooling are determined by the number of trials, tests, and levels. The effective degrees of freedom are the degrees of freedom, modified by the weighted average autocorrelation of lag 1 over the tests.

confidence limit.[58] The hypothesis of equal means of delays is tested by the F test at the 5% significance level. The trials cannot be considered to come from the same population because the F statistic is greater than the 5% point of the F distribution. However, the test means can be considered to come from the same population because the F statistic is less than the 5% point of the F distribution. The information from the hypothesis tests can be used in various ways depending upon the objective of the experiment. (Four of these ways are discussed in the four sections, 6.2 - 6.5.)

C. Shell Script Implementation

The identification of each test comprises a line in the file **log.acc** or **log.xfr**, depending upon whether access-disengagement tests or user information transfer tests are to be analyzed. The identification consists of the test number and the levels of each variable condition. The experimenter selects a level of a variable condition, say, **xxx**[59], then types

grep xxx log.acc > log.wrk

This UNIX[tm] utility copies the identification of all tests conducted at the level **xxx** into the file **log.wrk**.

Depending upon which type of performance parameter is to be analyzed, one of three shell scripts (i.e., **delay**, **rate**, and **fail**) can be activated to analyze the tests identified in **log.wrk**. For example, Access Time and User Fraction of Access Time can be analyzed from multiple tests by typing

delay ac i

where **i** denotes that pooling is to occur with respect to the **i**th variable condition in the list.

---

[58]The effective degrees of freedom for the confidence limits (not listed) are the degrees of freedom (determined by the sample size), modified by the average autocorrelation of lag 1 over the trials.

[59]In Figure 42, the variable condition is Source Site and the level is **A**. Hence, **xxx** is the character, **A**.

D. Operator Implementation

The operator can also analyze data from multiple tests by typing

star

and supplying the desired responses and data. However, the following restrictions apply:

- Delay data must be entered by files (one file for each test),
- failure probability data must be entered by keyboard, and
- rate data cannot be analyzed by an operator.

6.2 Estimate a Performance Parameter

Estimation consists of estimating the mean of a performance parameter and its confidence limits. The estimate can be obtained from a single test or from multiple tests.[60]

6.2.1 Single Test

All performance parameters except User Information Bit Transfer Rate and User Fraction of Input/Output Time can be estimated with desired precision from a single test. Because each test of these performance parameters results in at most one trial, they can be estimated with precision only from multiple tests.

Estimates and their confidence limits from single tests can be used in the equations for precision (Appendix A in Volume 5) to determine the achieved precision and compare it with the specified precision.

6.2.2 Multiple Tests

A performance parameter can be estimated from multiple tests for two purposes:

- More Precise Estimate. It can provide a more precise estimate of a performance parameter, often from tests conducted under

[60]Time parameters can be estimated more thoroughly than by estimates of their means and confidence limits: They can be estimated by histograms (sample densities) and box plots (abbreviated histograms) as shown in Section 7.

a single combination of levels (such multiple tests provide trials through replication).

- Representative Estimate. It can provide a single, representative estimate of a performance parameter from tests conducted under multiple combinations of levels.

In any case, the standard deviations of the primary performance parameter from each test should be somewhat the same - otherwise the trials of the tests probably do come from different populations, and an a priori condition of the hypothesis test has not been met.

A. More Precise Estimate

The experimenter selects a set of tests. Usually, the set consists of replications of tests conducted at the same combination of levels or at the same combination of levels - except one.

If the trials of each test can be pooled, no variable condition is a factor: The pooled estimate should be more precise than that provided by a single test.

If the trials cannot be pooled, there may be unknown variable conditions.

B. Representative Estimate

The experimenter selects a set of tests. Usually the set is selected so that, instead of many estimates (each for different combinations of levels), there will be a single estimate that is representative of the different combinations of levels.

If the trials of each test can be pooled, no variable condition is a factor. In this case, the trials have been shown to come from the same population, and the estimate from pooling is representative.

If the trials cannot be pooled, at least one variable condition is a factor. Analysis continues by testing whether the means of each test can be pooled. If so, they are pooled, and the next best estimate of the performance parameter has been obtained (i.e., precision is less than if the trials of tests could be pooled).

If the means of each test cannot be pooled, the means of each level of the selected variable condition are pooled. There is no significance test for this pooling, and precision is less than if the means of each test could be pooled.

6.3 Determine Acceptability of a Performance Parameter

An acceptance test is a hypothesis test that can determine whether the mean of a performance parameter equals or exceeds an acceptable (threshold) value. Hence, an acceptance test is appropriate to determine the acceptability of a performance parameter for at least two experiment objectives:

- Acceptance. An acceptance test can determine if the mean of a performance parameter is acceptable.
- Maintenance. An acceptance test can determine if the system requires maintenance to return the mean of a performance parameter to an acceptable value. (A quality control chart would be more appropriate if observations are acquired during operation.)

The following three concepts are required for acceptance testing:

- Threshold. The (threshold) acceptable value is specified. This is a value that can be accepted with indifference.
- Interval of Uncertainty. Because a sample has a finite number of trials, an interval of uncertainty exists about the threshold value. This interval is defined by two values, one that is considered to be totally satisfactory and one that is considered to be totally unsatisfactory. The narrower the interval of uncertainty, the greater the precision.
- Null Hypothesis. The null hypothesis states that the population value of the performance parameter is equal to the totally satisfactory value.[61] Because we are interested in whether the performance parameter value is better than the totally satisfactory value, this hypothesis is tested by a one-sided test.

Acceptance tests attempt to achieve two objectives with stated precision:

---

[61]It should be understood that a performance parameter value better than the totally satisfactory value is even more acceptable, so the composite hypothesis (better than or equal to) is not stated.

- Avoid Type I Error. The probability of rejecting a performance value that is totally satisfactory is to be $\alpha = 0.05$ or less (a probability called the significance level). This type of error is called a Type I error. The 5% significance level is traditionally used, but it could be, say, 1% if the loss incurred from committing this error would be large. The null hypothesis would be accepted at the α significance level if all or part of the $100(1 - 2\alpha)\%$ confidence interval of the performance parameter estimate lies in the totally satisfactory interval, and rejected otherwise. Since NTIA analysis uses 90% or 95% confidence limits, α should be 5% or 2.5%, respectively.[62]

- Avoid Type II Error. The probability of accepting a performance parameter value when its value is totally unsatisfactory is β. This type of error is called a Type II error. This probability is achieved by selecting a sufficiently large sample size (see Volume 2).

The probability of acceptance is some function of the performance parameter value, called the operating characteristic (OC). The concepts of acceptance testing are depicted by the schematic OC curve in Figure 43. In this figure,

- the probability of accepting the hypothesis when performance is totally satisfactory is $1 - \alpha$,

- the probability of accepting the hypothesis when performance is at the threshold value is 0.5, and

- the probability of (incorrectly) accepting the hypothesis when performance is totally unsatisfactory is β.

Confidence limits obtained from pooled data from multiple tests are not appropriate for acceptance tests.

---

[62]The $100\alpha\%$ significance level (i.e., one-sided) corresponds to a $100(1 - 2\alpha)\%$ confidence level (i.e., two-sided).

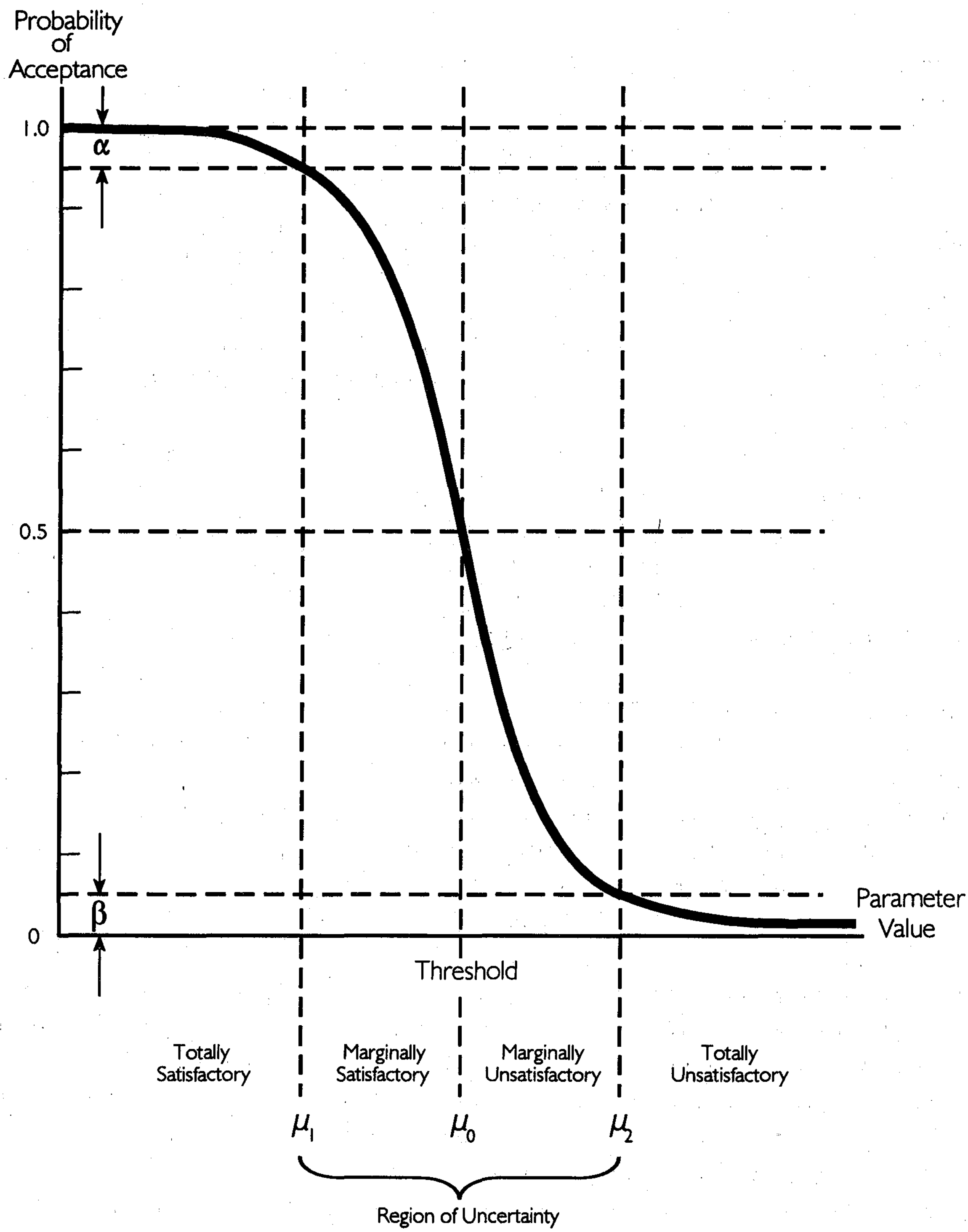


Figure 43. Schematic operating characteristic curve of a sampling plan for an acceptance test.

6.4 Compare a Performance Parameter from Two Services

Performance parameters from two services can be compared. The null hypothesis states that the means of the two performance parameters are equal:[63]

$$H_0: \mu_1 = \mu_2.$$

If estimates of the two means, each obtained under the same conditions, are significantly different at the α = 5% significance level, the performance parameter from one service is preferred.

Hypotheses are tested by hypothesis tests, and the appropriate hypothesis test depends upon whether the performance parameter is a time performance parameter or a failure probability performance parameter. In either case, program **star** can be used to compare a performance parameter from two tests, each conducted under the same combination of levels - except Network, of course.[64]

6.4.1 Time Performance Parameters

If the hypothesis test shows that the trials from the two tests can be combined, neither of the two services can be preferred for that performance parameter. If the trials from the two tests cannot be combined, one service can be preferred for that performance parameter. Choose the one whose estimate is preferred. For example, if Access Time is the tested performance parameter, choose the service with the lesser estimate of Access Time.

6.4.2 Failure Probability Performance Parameters

If the hypothesis test shows that the trials from the two tests can be combined, neither of the two services can be preferred for that performance parameter. If the trials from the two tests cannot be combined, choose the service having the smaller estimate of failure probability.

6.5 Determine if a Variable Condition Affects a Performance Parameter

Two methods from analysis of variance are available to determine if a variable condition is a factor for a performance parameter: linear regression

---

[63]This is a special case of the analysis of multiple tests in Section 6.5 (because only two tests are used). The purpose there is to determine if a variable condition is a factor, not to compare two performance parameter values.

[64]Since Network (i.e., service) is the only variable condition when comparing services, no discussion of the implications of selecting a variable condition is needed - as it is for estimating (Section 6.2) and for determining if a variable condition is a factor (Section 6.5).

analysis (for primary time performance parameters) and hypothesis tests of the null hypothesis of equal means of tests (for any performance parameter).

6.5.1 Linear Regression Analysis

Linear regression analysis can determine if a certain variable condition (having quantifiable levels) is a factor for a primary time performance parameter. The values of this primary time performance parameter (i.e., trial values or estimates from more than one trial) can be plotted at the measured levels of the variable condition. The levels of the variable condition are values of the independent variable, and the performance parameter values are values of the dependent variable. In the absence of measurement error, these points would lie on a curve. For simplicity, assume the curve is a straight line. The straight line is determined by the method of least squares: It is the line that minimizes the sum of the squares of the vertical distances between the points and the line.

The slope, b, of the regression line (called the regression coefficient) indicates the degree to which the performance parameter depends upon the variable condition; a slope of zero suggests no dependence, in which case it is concluded that the variable condition is not a factor for that performance parameter.

We can test the null hypothesis that $b = b'$ where b' is some specified slope. In our case, we want to determine if a variable condition is a factor for a performance parameter, so we would test the hypothesis that the slope is zero (i.e., $b = 0$). That is,

$$H_0: b = 0.$$

Volume 6 of this report shows how regression lines can be plotted for primary time performance parameters. However, no additional analysis is provided.

6.5.2 Hypothesis Tests

This analysis of multiple tests assumes that the experiment is designed to investigate the effect of a <u>single</u> variable condition upon a performance parameter. Hence, the design can be considered to be either a completely randomized design or a randomized block design (each with one variable condition). Data from tests conducted under different levels of this variable condition are pooled.

The experiment may have been designed to determine if one or all of N identified variable conditions are factors. To determine if they are, one or more tests have probably been conducted at different combinations of levels of the variable conditions. Each combination of levels of the variable conditions defines a population, and the null hypothesis states that the means of the populations, say k of them, are equal:

$$H_0: \mu_1 = \ldots = \mu_k.$$

The null hypothesis is tested by a hypothesis test. The statistic is compared with an appropriate distribution at, say, the α = 5% point. If the value of the statistic is less than this value, the null hypothesis is accepted, and the tested variable conditions are not factors. Otherwise, at least one is.

Program **star** can determine if <u>any</u> of the variable conditions are factors.[65] Suppose a set of tests having j variable conditions are to be analyzed (i.e., $1 \leq j \leq N$). Tests from the experiment can be selected for two purposes:

- <u>Determine if One Variable Condition is a Factor</u>. Select tests having the same combination of levels except for those of one variable condition (i.e., j = 1). The hypothesis test determines if that variable condition is a factor.

- <u>Determine if at Least One of j Variable Conditions is a Factor</u>. Select tests having the same combination of levels except for those of j variable conditions (i.e., j > 1). The hypothesis test determines if <u>any</u> of those j variable conditions are a factor. However, it will not indicate which are. Of the j variable conditions, one must be selected to test the null hypothesis that the means of its levels are equal. Selection of the variable condition has the following three effects:

    - <u>Pooling Among Trials</u>: Selection does not affect the acceptance of the null hypothesis that test means are equal nor does it affect the estimates of the mean of the trials and its confidence limits.

    - <u>Pooling Among Test Means</u>: Selection affects the acceptance of the null hypothesis that level means are equal, but it does not affect the estimates of the mean of test means and its confidence limits.

    - <u>Pooling Among Level Means</u>: There is no null hypothesis, but selection affects the estimates of the mean of level means and its confidence limits.

---

[65]Methods to define statistics that test whether <u>multiple</u> variable conditions are factors can be found in many experiment design texts.

Example: Eleven tests of Access Time have been conducted using Service A. The five identified variable conditions are

- Source Site (three levels: Fort Worth, Seattle, and Washington D.C.),
- Day of Week (five levels: Monday, Tuesday, Wednesday, Thursday, and Friday),
- Time of Day (six levels, each containing 4 hours: identified by 1, 2, 3, 4, 5, and 6),
- Interaccess Delay (One level: 55 s), and
- Destination Site (One level: Boulder).

Since two of the variable conditions have only one level, they can be considered to be fixed conditions. Therefore, there are really only three variable conditions. Determine if any of these three variable conditions are factors for Access Time for Service A.

Solution: Suppose the identification of these 11 tests has been stored in the file **log.wrk**. Then type

delay ac 1

where variable condition 1 has been arbitrarily selected (i.e., Source Site); its selection will not affect acceptance of the null hypothesis that the test means are equal. Program **star** is executed by the shell script **delay**, and the results are shown in Figure 44. (Refer to Section 6.1.2 for an explanation of this type of figure.) Since the * does not appear in the **among trials** row, at least one of the three variable conditions is a factor for the performance parameter, Access Time.

Analysis of Multiple Tests

Access Time
Variable Condition 1

Thu Jan 26 14:06:14 MST 1989

Test	--- Variable Conditions ---	Trials	---- Times ---- Mean	Std Dev	User Fractions Mean	Std Dev
775	ftw netA fri 1 A55 bol	20	38.291	1.608	0.0397	0.0199
823	sea netA fri 2 A55 bol	20	42.439	1.527	0.0339	0.0047
815	sea netA fri 6 A55 bol	20	41.576	1.269	0.0352	0.0044
835	sea netA mon 3 A55 bol	15	42.954	1.325	0.0345	0.0053
858	sea netA thu 1 A55 bol	20	42.284	1.338	0.0345	0.0053
876	sea netA thu 4 A55 bol	20	42.313	2.197	0.0350	0.0064
811	sea netA thu 5 A55 bol	19	41.163	1.015	0.0373	0.0065
997	wdc netA thu 3 A55 bol	17	41.751	2.198	0.0356	0.0075
928	wdc netA tue 1 A55 bol	20	44.500	4.380	0.0332	0.0068
952	wdc netA tue 5 A55 bol	20	39.813	1.625	0.0368	0.0043
978	wdc netA wed 4 A55 bol	18	42.304	1.820	0.0351	0.0054

TIMES (W) AND FRACTION OF TIMES (V)

NUMBER OF TRIALS = 209
NUMBER OF TESTS = 11
NUMBER OF LEVELS = 3

WEIGHTED AVERAGE AUTOCORRELATION COEFFICIENT
OF LAG 1 OVER THE 11 TESTS = .3927E+00 #
AVERAGE AUTOCORRELATION COEFFICIENT
OF LAG 1 OVER THE 209 TRIALS = .4998E+00 @

	EFFECTIVE DEGREES OF FREEDOM		F STAT.	F DIST. (5%)		95% LOWER CONFIDENCE LIMIT	ESTIMATE OF THE MEAN	95% UPPER CONFIDENCE LIMIT	
AMONG TRIALS	80	10	.4961E+01	.1963E+01	W	.4112E+02	.4173E+02	.4234E+02	
	-	-	-	-	V	.3423E-01	.3547E-01	.3671E-01	
AMONG TESTS	8	2	.4011E+01	.4460E+01	W	.4067E+02	.4176E+02	.4286E+02	*
	-	-	-	-	V	.3416E-01	.3551E-01	.3686E-01	
AMONG LEVELS	-	-	-	-	W	.3536E+02	.4083E+02	.4631E+02	
	-	-	-	-	V	.2953E-01	.3676E-01	.4417E-01	

USED TO DETERMINE THE EFFECTIVE DEGREES OF FREEDOM FOR THE F TEST.
@ USED TO DETERMINE THE EFFECTIVE DEGREES OF FREEDOM FOR THE CONFIDENCE LIMITS.
* AT RIGHT OF UPPER CONFIDENCE LIMIT INDICTES THIS POOLING
IS ACCEPTABLE AT THE 5% LEVEL.

Figure 44. **Example of a test of variable conditions for a time performance parameter.**

7. DISPLAY THE DATA

7.1 Introduction

Graphics software is an NTIA enhancement of the ANSI specifications. It provides four types of two-dimensional plots: histograms, box plots, chronological plots, and regression lines. The performance parameters can utilize these plots according to the type of random variable:

- Delays. The four primary delay performance parameters (i.e., Access Time, Block Transfer Time, Source Disengagement Time, and Destination Disengagement Time) can be analyzed by any of the four types of plots.

- Rate. The primary rate performance parameter, User Information Bit Transfer Rate, can be analyzed by all but chronological plots. However, since there is at most one value per test for this performance parameter, values from multiple tests would be required.

- Failures. Failure probability performance parameters can't be analyzed by these plots

Table 15 lists the available plots for the primary delay performance parameters. It shows the number of plots per page and whether data from single or multiple tests may be plotted.

The programs described here are designed to be as general as possible, so that the user can generate plots from any file of numerical data. This software is designed to use an HP7475A vector plotter, but it can use other plotters after appropriate modifications.

Table 15. Available Plots for Primary Delay Performance Parameters

		PLOTS/PAGE	
		Single Plot	Multiple Plots
HISTOGRAMS	Single Tests	√	-
	Multiple Tests	√	-
BOX PLOTS	Single Tests	√	√
	Multiple Tests	√	√
CHRONOLOGICAL PLOTS	Single Tests	√	√
	Multiple Tests	-	-
REGRESSION PLOTS	Single Tests	√	√
	Multiple Tests	√	√

7.2 Draw Histograms

A histogram is an estimate of a probability density function. It is created from outcomes in a sample from a population. The outcomes are ordered, the abscissa is partitioned judiciously, and the percent of outcomes within each segment of the partitioned abscissa is determined. The histogram is created by plotting these percentages (or frequencies) at the midpoint of each segment.[66]

Figure 45 is an example of a histogram of 81 pooled trials of Source Disengagement Time for a public data network (PDN).

7.3 Draw Box Plots

A box plot is a consolidated histogram. The outcomes of a sample are ordered. Then the following values are determined; the minimum, the 25th percentile, the 50th percentile (i.e., the median), the 75th percentile, and the maximum. All five values are indicated by a tic mark on a line segment; further,

[66]The shape of the histogram can suggest the population. Then a chi-squared goodness of fit test can determine if there is a significant difference between the sample (histogram) and the suggested population (density).

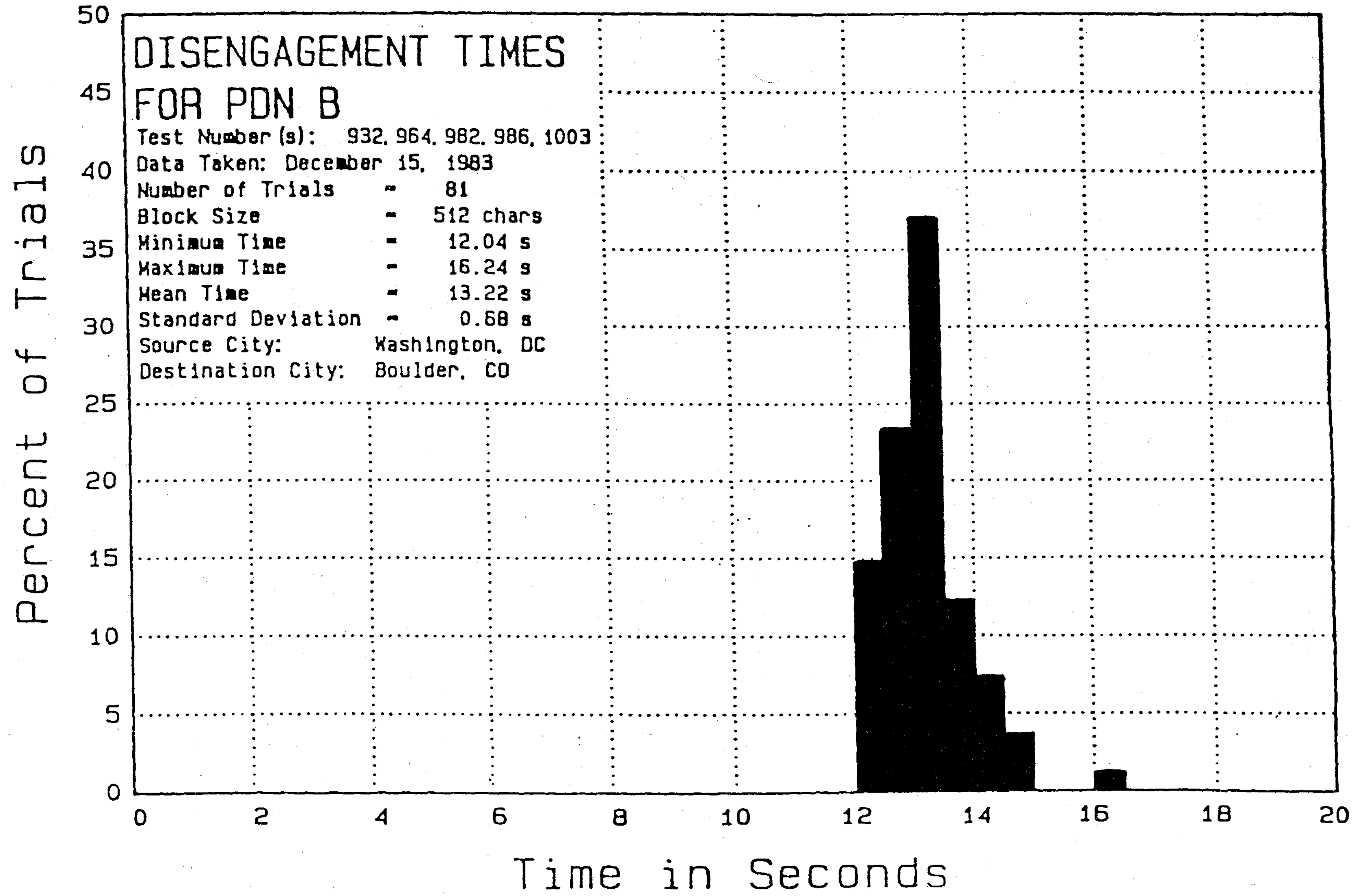


Figure 45. Example of histogram of Source Disengagement Time.

the tic marks at the 25th and 75th percentiles form the ends of a rectangle. This rectangle emphasizes the middle 50% of the outcomes. Since box plots are narrower than histograms, box plots of multiple samples can be plotted side-by-side and compared.

Figure 46 is an example of three sets of box plots of Block Transfer Times for different Block Sizes. Within each set are box plots of five networks, each containing box plots of transmissions from five remote sites and with a different utilization. The plots show that Block Transfer Time is less and less variable over the direct dial networks "D" and "F" than over the three PDNs ("A", "B", and "C"). The effect of Utilization can be observed by comparing, within each cluster of networks, the box plots on the right (high utilization) with those on the left (low utilization).

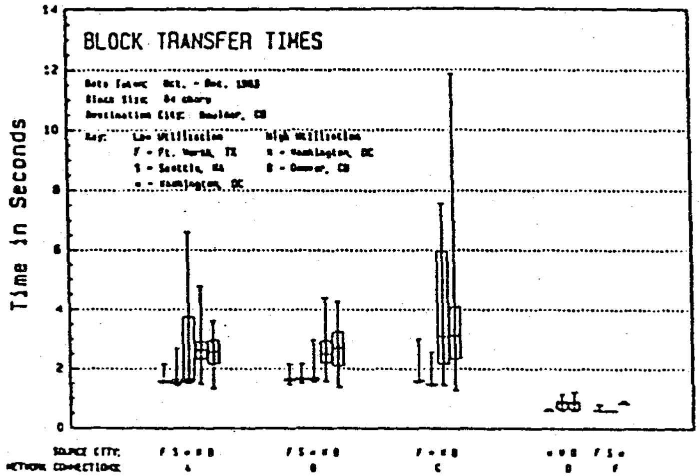
7.4 Draw Chronological Plots

A chronological plot is a plot of the outcomes from a sample in the order in which they occurred. This plot is useful to spot trends as a function of time. It also shows dependence that may exist between trials; dependence can seriously affect the precision of an estimate, decreasing it if the autocorrelation is positive.

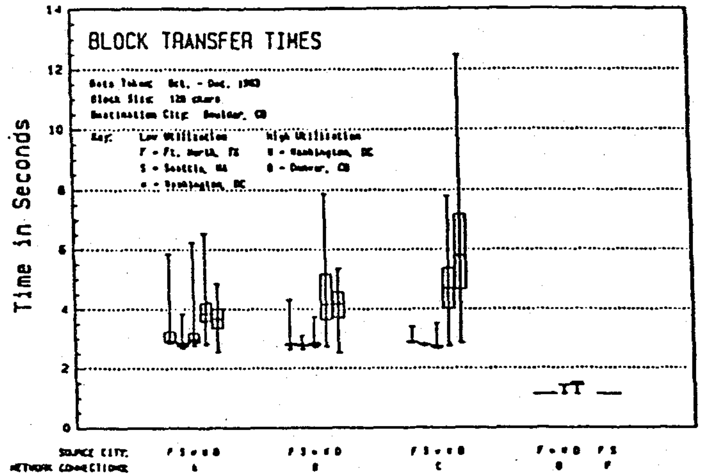
Figure 47 is an example of two sets of chronological plots of Block Transfer Times for 128 character blocks. Each set represents a different remote site (e.g., Denver and Washington, DC). Within each set are chronological plots over three PDNs (i.e., "A", "B", and "C") and a direct dial network (i.e., "D"). Block Transfer Times for Network D are essentially constant, whereas those of PDNs vary considerably; these networks use high utilization.

7.5 Draw Regression Plots

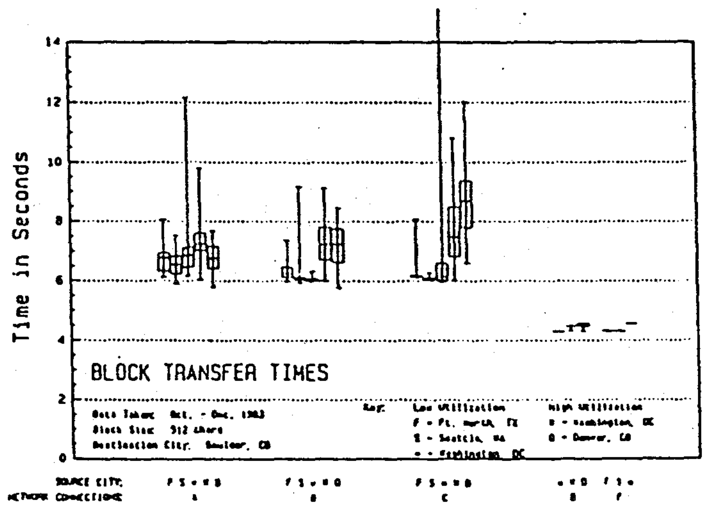
Outcomes may be affected by one or more variable conditions that exist during a test. Regression is the process of estimating the outcome (i.e., dependent variable) from the values of a quantity (i.e., independent variable) in a sample. One of the main purposes of regression is curve fitting. NTIA software fits a straight line to data by using the method of least squares. This method selects the line that renders a minimum, the sum of the squares of the



a. Short (64—character) Blocks

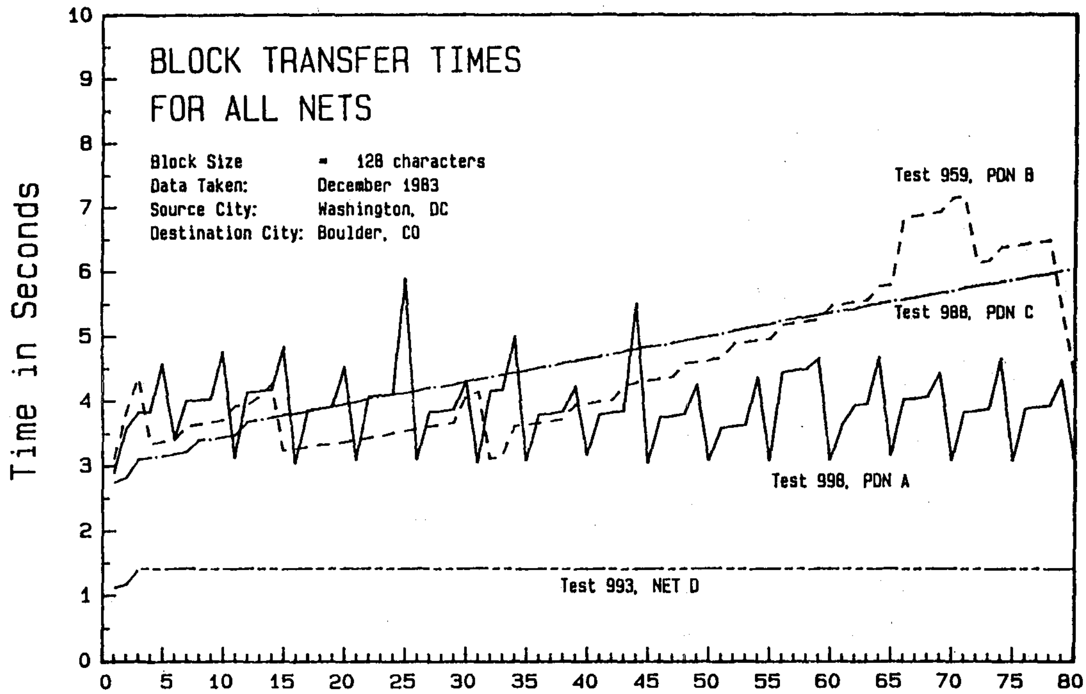


b. Medium (128—character) Blocks



c. Long (512—character) Blocks

Figure 46. Example of box plots of Block Transfer Times.



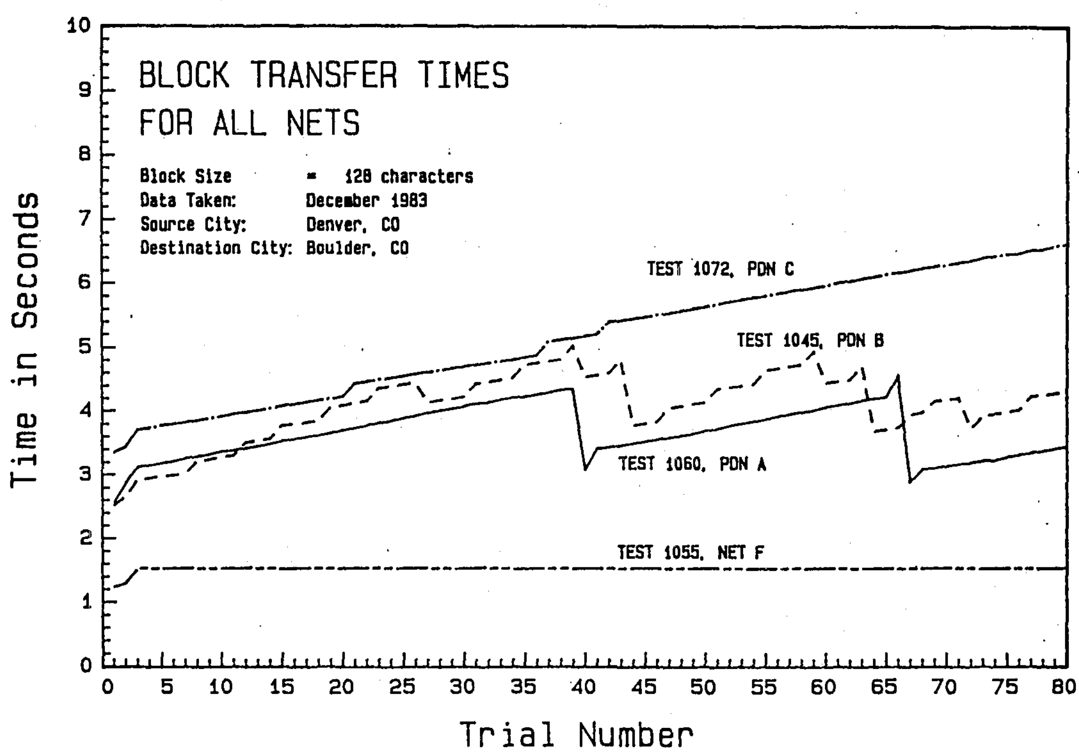


Figure 47. Example of chronological plots of Block Transfer Times.

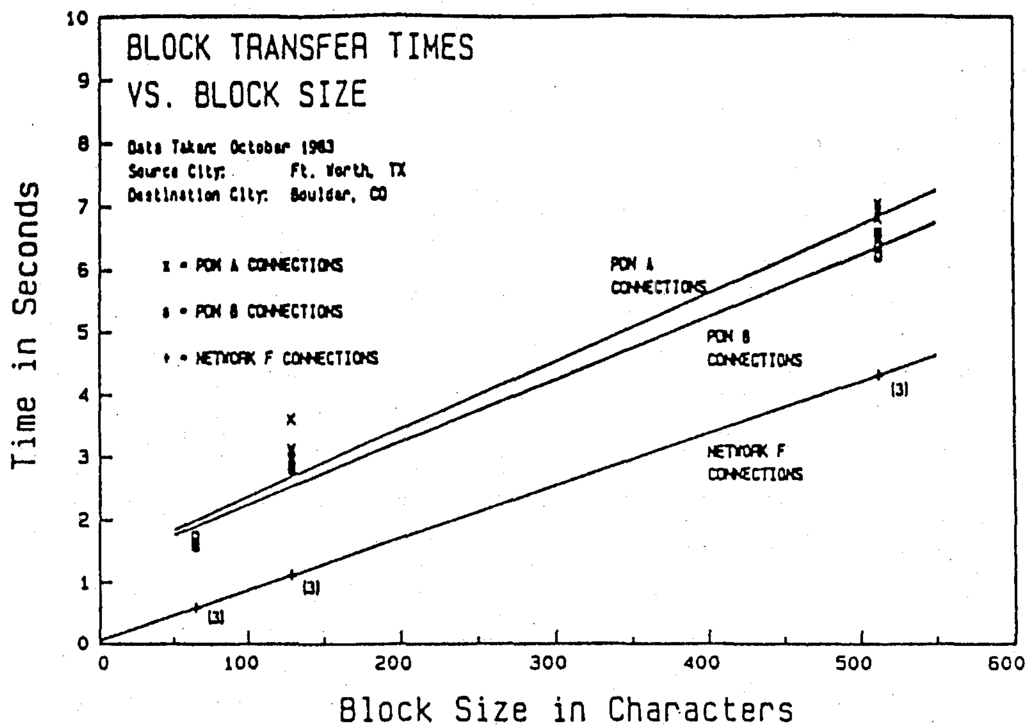
vertical distances between each outcome and the corresponding point on the line. The slope of a regression line indicates the degree of dependence of the performance parameter upon the different levels of a variable condition. For example, it would show the effect of Block Length upon Block Transfer Time.

A regression plot consists of data plotted on the X-Y plane and the regression line. Figure 48 is an example of two regression plots of Block Transfer Time versus Block Size. They are grouped according to Site (i.e., Fort Worth and Seattle). The lines represent transmissions over two PDNs (i.e., "A" and "B") and the direct dial network (i.e., "F"). In all cases the performance parameter is affected by Block Length. Transmission time over the direct dial networks is shorter because PDNs have "store and forward" switches which cause delays. Transmission time for network F is nearly equal to that of the access line speed of 128 characters/second.

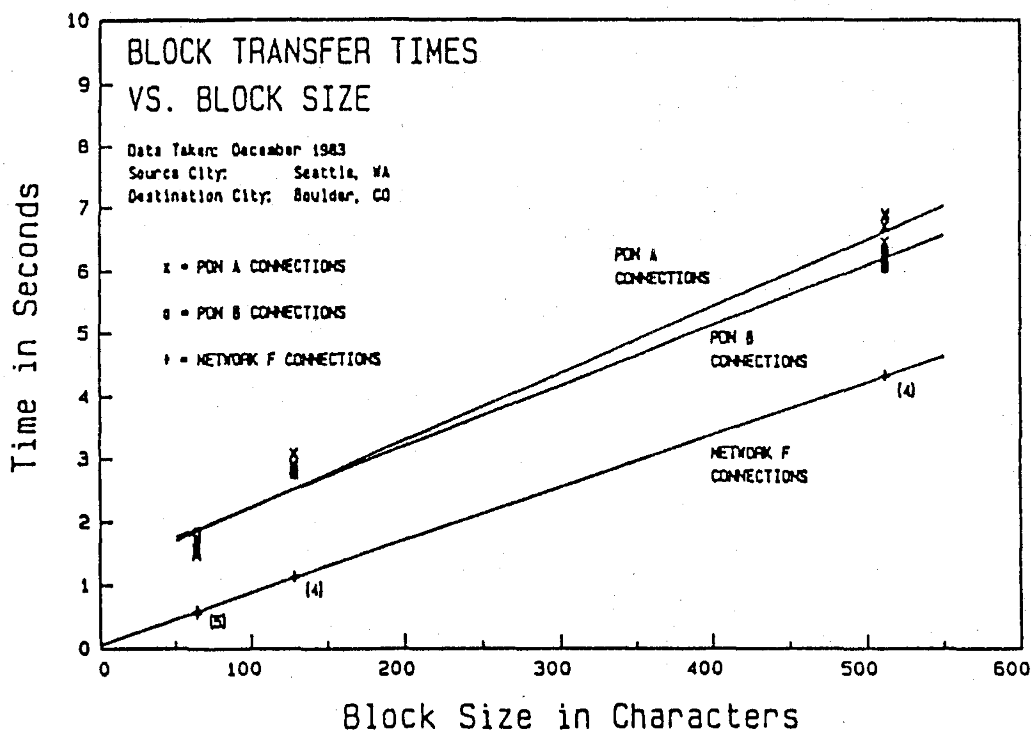
7.6 Create Box, Chronological, and Regression Plots on the HP7475A

Box, chronological, and regression plots require the same preparation:

- <u>Prepare and Plot the Axes</u>. Create the menu file, create the legend file, and type, **vpaxes menufile**.
- <u>Plot the Graph</u>. Select the line types and the medium.
- <u>Plot the Legend</u>. Use the routine called **movit**.



a. Ft. Worth, TX Results



b. Seattle, WA Results

Figure 48. Example of regression plots of Block Transfer Time.

PART III. NTIA CODE

The version 2.1 of the NTIA software code consists of many programs, subprograms, and shell scripts that are written in FORTRAN 77 and C programming languages (Kernighan and Richie, 1978). Shell scripts in both the AT&T Bourne shell and the Berkeley C shell link most of the software into an automatic process on each PC. The source code is not reproduced in this report but is contained in the set of seven PC-DOS^{tm} and MS-DOS^{tm} 5-1/4" diskettes whose numbers correspond to the volumes of this report.[67]

The software is designed to operate under the Microport System V.3 operating system.

Part III describes the diskettes that contain the code, the installation procedure, and the directory structure. Beginning in Part III of this volume and continuing throughout this six volume report, actions required of the reader are discussed in shaded panels.

The diskettes can be obtained from

Performance Measurement Software
U.S. Department of Commerce
NTIA/ITS.N3
325 Broadway
Boulder, CO 80303-3328

---

[67]PC-DOS is a trademark of International Business Machines, Inc, and MS-DOS is a trademark of Microsoft, Inc.

8. DISKETTES

8.1 Software Diskettes

The seven software diskettes are:

- Diskette #1 (Sample Data Files). This diskette contains ASCII text files of sample data and some documentation.
- Diskette #2 (Experiment Design). The code on this diskette is both FORTRAN 77 and C language. The FORTRAN program **star**, that can be used to determine the sample size, is the same as that on diskette #5 and is not included on this diskette.
- Diskette #3a (Data Extraction). The code on this diskette is C language. It is system-dependent because it uses system calls that may be unique under the current vendor's software.[68]
- Diskette #3b (Data Conversion). The code on this diskette is C language.
- Diskette #4 (Data Reduction). The code on this diskette is FORTRAN 77. Due to the complexity of the code, it is recommended that it be run on a fairly fast CPU.
- Diskette #5 (Data Analysis). The code on this diskette is FORTRAN 77 (i.e., program **star**) except for one C program.
- Diskette #6 (Data Display). The code on this diskette is C language which must be modified if neither the HP7475A vector plotter nor an emulator is not used.

8.2 Demonstration Diskettes

A companion set of five demonstration diskettes is provided to operate in MS-DOS[tm] or PC-DOS[tm]. Each of the major software modules (except those of data extraction) is included on a demonstration diskette along with a batch file and sample data. The five diskettes and their modules are:

- Diskette #2D (Experiment Design): **star**
- Diskette #3bD (Data Conversion): **merge** and **reform**

[68]Future versions will conform to the ANSI C Language Standard and the IEEE POSIX Standard when vendors provide this capability, thus providing portability of the software to other operating systems.

- Diskette #4D (Data Reduction): **prolog**, **analyz**, and **epilog**
- Diskette #5D (Data Analysis): **star**
- Diskette #6D (Data Display): **vp**, **vpaxes**, **vplegend**, **histogrm**, **box**, and **hpgraph**

Because this report is quite extensive, instructions for using the demonstration diskettes can be obtained as follows:

> To use a demonstration diskette, place it in a personal computer having a math coprocessor chip, look in a **README** file for the names of the demonstrations, and type each of the names.

9. SOFTWARE INSTALLATION

Software from the seven disks must be installed on at least one computer designed to process tests. Since the software is distributed on DOS-formatted floppy disks, the user needs to read and convert DOS files to UNIX^{tm}. Many system have a **doscp** (i.e., DOS copy) command to do this.

The software should compile and execute on most UNIX systems compatible with the SVR3 interface. However, if there are incompatibilities, change the makefiles, the install script, or the source code. Also, check any **README** files that accompany your version of the software.

The following instructions assume the software is installed under Microport System V.3 operating system.

At a minimum, the extraction software from the seven diskettes must be installed in both the source and destination computers. In order to install the software, the UNIX^{tm} system must have a **/dev/dos** directory that contains disks linked to the proper I/O devices. Specifically,

- disk drive A must be linked to a high density (1.2mb) I/O device,
- disk drive B must be linked to a low density (360kb) I/O device, and
- hard disk C must be partitioned for DOS^{tm}.

Each computer should have a Microport System V.3 operating system (version 3.0e or later) with a FORTRAN 77 compiler.[69]

[69]If another operating system is used, the command **doscp** must be replaced with an appropriate command and **install** must be edited. If the FORTRAN 77 compiler is not available, either create a C shell F77 to invoke your compiler or alias your F77 compiler.

To install the software, follow these steps:

1. Have the superuser create a new account called **net** with home directory **/usr/net**.

2. Log in with user name.

3. Insert diskette #1 in drive B.

4. Type **doscp B: install.** and press return.

5. Type **sh install**. Follow these instructions from the screen:

   - log out by typing **logout** or **^D**,
   - have the superuser log in,
   - have the superuser change the directory to **net's** home directory,
   - have the superuser type **Rootdir**, and
   - have the superuser log out by typing **logout** or **^D**.

6. Log in by typing **net**

7. Type **sh install**. Now **install** requests that diskettes #2 - #6 be inserted. When **install** is finished, the message **install complete** appears on the screen.

8. Type **make all**. This command compiles the programs by invoking **make** in each directory. The procedure may require several minutes.

9. Type **make move**. This command moves the compiled programs to directory **~net/bin**.

10. Type **make clean**. This command removes unnecessary files created by **make**.

11. Type **rehash**. This command renders the programs available.

Installation is now complete, and the files have been placed in the appropriate directories.

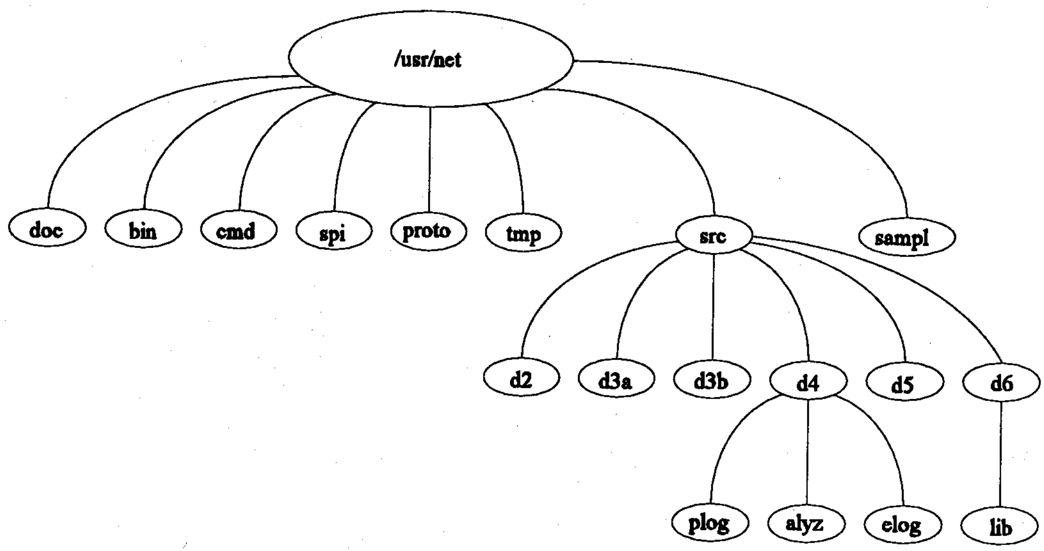
10. DIRECTORIES

The NTIA software has been placed in two directories, the **/usr/net** directory and the **/usr/data** directory. The first directory is the home directory (i.e., the directory in which you are placed when you log on). It contains the software necessary to process a test. The second directory will contain data from the experiment. The structures of these directories are shown in Figure 49.

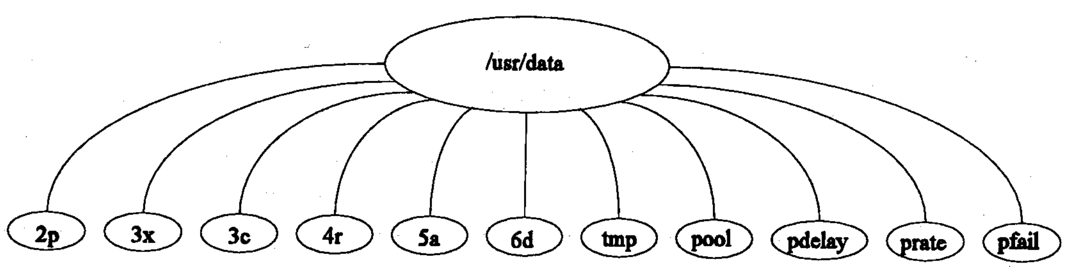
10.1 Directory /usr/net

Log on with user name. This directory contains programs, shell scripts, and files necessary to process each test. It has eight subdirectories:

- **doc**: software documentation
- **bin**: compiled programs
- **cmd**: shell script files
- **spi**: specified performance values files
- **proto**: protocol files
- **tmp**: temporary (scratch) files
- **src**: diskettes 2-6. Each diskette should have an **all** option. This subdirectory contains six other subdirectories of programs and shell scripts:
    - **d2**: source code for experiment design FORTRAN 77 program
    - **d3a**: source code for data extraction (on-line) C programs
    - **d3b**: source code for data conversion (off-line) C programs
    - **d4**: source code for data reduction FORTRAN 77 programs
        - **plog**: program to check for errors and consolidate specifications and overhead information
        - **alyz**: program to identify next trial and determine outcome



a. Structure of the net directory.



b. Structure of the data directory.

Figure 49. Directory structure of NTIA code.

- **elog**: program to format results
- **d5**: source code for data analysis FORTRAN program (i.e., **star**)
- **d6**: source code for data display C programs
- **lib**: library C routines for data display

- **sampl**: sample data files

Figure 50 is a list of files contained in each subdirectory of **/usr/net**.

10.2 Directory /usr/data

This directory contains data from the tests. It has eleven subdirectories:

- **2p**: preliminary characterization test data
- **3x**: extracted binary data (extracted on-line)
- **3c**: converted ASCII data (converted off-line)
- **4r**: reduced ASCII data (i.e., outcomes)
- **5a**: analyzed data (i.e., measurement result summary)
- **6d**: displayed data (i.e., "quick look")
- **tmp**: working directory for post-test processing
- **pool**: working directory for pooling
- **pdelay**: delay performance data files for pooling (e.g., **nnnnnac**).
- **prate**: User Information Bit Transfer Rate and User Fraction of Input/Output Time performance data file for pooling (i.e., **thrput**)
- **pfail**: failure performance data file for pooling (e.g., **failac**).

The list of files in **/usr/data** is too lengthy to list in a figure.

```
Makedir        cmd/           install.doc    preface.doc    spi/
Makefile       codes          log            preface.r      src/
Rootdir        data.x         log.bak        preface.x      sys/
bin/           doc/           net-vm96.doc   proto/         tmp/
clkcal         install        netcodes       sampl/

./bin:
acctime*   gentem*    mkpre*     recv*      show-o*    tablt-a*   vp*
analyz*    histogrm*  mkvpleg*   reform*    st*        tablt-x*   vpaxes*
batchid*   hpgraph*   movit*     relate*    star*      truetime*  vplegend*
calibr*    merge*     prolog*    settime*   swappit*   tt*        vpz*
epilog*    mkdate*    qklog*     show-h*    tablf-a*   tweak-h*   xfertime*
firsttwo*  mklog*     quartile*  show-hd*   tablf-x*   tweak-o*   xmit_net*

./cmd:
cksum*       fail*        mkhist*      pretabt*     reduc*       runxt.tst*
clear*       fail-a*      mkinfo*      prmt-a*      reduc-a*     runxtf*
clkcal*      fail-x*      mktab*       prmt-x*      reduc-x*     runxtf.tst*
conver*      faildev*     mktem*       qkhist*      repeat*      time-a*
delay*       fileck*      mkvp*        qklook*      runr*        time-x*
display*     head*        move.rcv*    qktab*       runumb*      tweakall*
do*          junk         move.xmt*    qkvp*        runx*        tweaknon*
dopre*       mean-dev*    mover*       qp*          runx.tst*
doqik*       meanrdev*    movex*       quarter*     runxf*
envvar*      mkasci*      pretabf*     rate*        runxt*

./doc:
cover.doc      cshrc          install.doc    logout         promotn.doc
cover1.doc     info-d1.doc    kermrc         profile        roadmap.doc

./proto:
net-bak.bol    net-vm96.bol   net-xs96.bol   xmt-vm19.bol   xmt_xs96.bol
net-vm19.bol   net-vm96.doc   old            xmt-vm96.bol

./sampl:
1424.chr       1424info.or    1424xfr.tab    cor            scr
1424.leg       1424info.ort   b1fail         csp            soi
1424.men       1424info.ox    b1o            data.x         spi
1424.tem       1424info.oxt   b2fail         doi            spi.acd
1424data.r     1424log.r      b2o            dui            spi.xfr
1424info       1424log.x      b3fail         log            sts
1424info.hr    1424logn.x     b3o            log.wrk        sui
1424info.hrt   1424mrs.fpr    b4o            par            sum
1424info.hx    1424mrs.tim    clkdata.cal    preface.r      thrput
1424info.hxt   1424xfr.his    coi            preface.x

./spi:
spi        spi.acd    spi.xfr

./src:
d2/    d3a/   d3b/   d4/    d5/    d6/
```

Figure 50 (Part 1). List of files in each subdirectory of usr/net.

```
./src/d2:
Makefile   mkpre.c

./src/d3a:
Makefile       clockusr.h     makefile.mp    pc.c           show-o.c
calibr.c       comsubs.c      makefile.pc    pdn_test.h     truetime.c
clksubic.c     connect.c      makefile.sco   recv.c         tty.c
clksubs.c      files.dir      maketime       settime.c      xmit_net.c
clock.c        io.h           mklog.c        show-h.c
clock.h        jtom.c         parsecmd.c     show-hd.c

./src/d3b:
Makefile      firsttwo.c    reform.c      tablf-a.c     tablt-x.c     xfertime.c
acctime.c     merge.c       relate.c      tablf-x.c     tweak-h.c
batchid.c     mkdate.c      swappit.c     tablt-a.c     tweak-o.c

./src/d4:
Makefile   alyz/      elog/      plog/

./src/d4/alyz:
Makefile   analyz.f   bitcor.f   dshift.f   exsbit.f   rdspec.f   ttsamp.f
access.f   ancill.f   catgap.f   dwload.f   exsrec.f   sshift.f   uiotim.f
aliput.f   assess.f   compar.f   exdbit.f   instat.f   swload.f   wrpars.f
aloput.f   bitcop.f   diseng.f   exdrec.f   orstat.f   transf.f   wrstat.f

./src/d4/elog:
Makefile   dattim.f   efchar.f   rdcsum.f   rdstat.f
aspref.f   dfchar.f   eliput.f   rdpars.f   summry.f
asumry.f   dsumry.f   epilog.f   rdspec.f   tsumry.f

./src/d4/plog:
Makefile   ckspec.f   conrsp.f   datxam.f
ckinfo.f   conrev.f   datcon.f   prolog.f

./src/d5:
Makefile    checkx.f    enterx.f    limit.f     qklog.c     xfile.f
anzflr.f    chisq-r.f   fdist-r.f   main.f      ssdflr.f    yfile.f
anztim.f    entera.f    ftest-r.f   multip.f    ssdtim.f    zfile-r.f
checki.f    enteri.f    level.f     poiss.f     studnt.f

./src/d6:
Makefile      histogrm.c    lib/          quartile.c    vplegend.c
box.c         hpgraph.c     mkvpleg.c     vp.c          vpz.c
gentem.c      junk          movit.c       vpaxes.c

./src/d6/lib:
arc.c         comment       exitplot.c    plot.h        reldir.c      speed.c
axis.c        daset.c       fill.c        plotchar.c    relsize.c
circle.c      dir.c         initplot.c    pmove.c       rmove.c
cirflat.c     dsset.c       label.c       poiflat.c     scale.c
clabel.c      ellipse.c     line.c        rect.c        size.c
commands      execplot.c    pen.c         rectang.c     slant.c
```

Figure 50 (Part 2). List of files in each subdirectory of usr/net.

```
./sys:
print*   turn*

./tmp:
2183.cor     2186.sum     2222db2.chr  2222db2.mnu   vol3.txt
2183.sum     2218info     2222db2.his  2222db2.tem
2186.cor     2221info     2222db2.leg  C2

./uw:
aces-tes  betime*   myfile    uope
atest.c   betime.c  run       uope.c
```

Figure 50 (Part 3). **List of files in each subdirectory of usr/net.**

11. REFERENCES

ANSI (1987), American National Standard for Information Systems - Data communication systems and services - measurement methods for user-oriented performance evaluation, ANSI X3.141-1987 (American National Standards Institute, Inc., New York, NY).

ANSI (1983), American National Standard for Information Systems - Data communication systems and services - user-oriented performance parameters, ANSI X3.102-1983 (American National Standards Institute, Inc., New York, NY).

ANSI (1992), American National Standard for Information Systems - Data communication systems and services - user-oriented performance parameters, ANSI X3.102-1992 (American National Standards Institute, Inc., New York, NY).

Cass, R.D. and M.J. Miles (1990), Design of experiments to quantify communication satellite system performance, NTIA report 90-269, November (NTIS Order Number PB 91-129635).

Crow, E.L., and M.J. Miles (1977), Confidence limits for digital error rates from dependent transmissions, Office of Telecommunications Report 77-118 (NTIS Order Number PB 267777).

Kernighan, B.W., and D.M. Richie (1978), *The C Programming Language* (Prentice-Hall, Inc., Englewood Cliffs, NJ).

Miles, M.J. (1984), Sample size and precision in communication performance measurements, NTIA Report 84-153 (NTIS Order Number PB 85-114270).

Miles, M.J., and E.L. Crow, *Statistical Methods for Data Communication Performance Evaluation* (Book to be published).

Seitz, N.B., D.R. Wortendyke, and K.P. Spies (1983), User-oriented performance measurements on the ARPANET, *IEEE Communications Magazine*, August, pp. 28-44.

Spies, K.P., D.R. Wortendyke, E.L. Crow, M.J. Miles, E.A. Quincy, and N.B. Seitz (1988), User-oriented performance communication services: measurement design, conduct, and results, NTIA Report 88-238, August (NTIS Order Number PB 89-117519/AS).

Wortendyke, D.R. and T.W. Butler (1991), Measurement of data performance parameters for data/VTC networks, IEEE Pacific Rim Conference, May 9-10, Victoria, B. C., Canada.

Wortendyke, D.R., M.H. Morris, and D.J. Atkinson (1989), Performance testing of wide area networks, University Programs in Computer-Aided Engineering, Design, and Manufacturing, ASCE Seventh Annual Conference, July 23-26, Laramie, WY.

Wortendyke, D.R., M.H. Morris, and D.J. Atkinson (1989), Performance testing of wide area networks, University Programs in Computer-Aided Engineering, Design, and Manufacturing, ASCE Seventh Annual Conference, July 23-26, Laramie, WY, pp. 311-317.

Wortendyke, D.R., N.B. Seitz, K.P. Spies, E.L. Crow, and D.S. Grubb (1982), User-oriented performance measurements on the ARPANET: Testing of a proposed standard, NTIA Report 82-112, November, 293 pp. (NTIS Order Number PB 83-159947).

APPENDIX: BIT AND BLOCK TRANSFER OUTCOMES AS DEFINED BY ANS X3.102

This appendix paraphrases the discussion of user information bit transfer outcomes and block transfer outcomes from the revised version of ANS X3.102. It is included here because it is considered too lengthy for Section 1.3.2.

Before defining either user information bit transfer outcomes or user information block transfer outcomes, it is convenient to define two dispositions of user information bits:

- Undelivered User Information Bit. A user information bit transferred from the source user to the system is called an undelivered source user information bit if it does not correspond to a user information bit transferred to the destination user. Undelivered user information bits can occur if the End of Block Transfer reference event does not occur.

- Extra User Information Bit. An extra (destination) user information bit is a user information bit transferred from the system to the destination user for which there is no corresponding user information bit transferred from the source user to the system. Extra user information bits can occur if the Start of Block Transfer reference event does not occur.

Undelivered user information bits are used to define the Lost Bit outcome and the Incorrect Block outcome. Extra user information bits are used to define the Misdelivered Bit outcome, the Extra Bit outcome, the Extra Block outcome, and the Misdelivered Block outcome.

A.1. Bit Transfer Outcomes

The six possible user information bit transfer outcomes are:

- Successful Bit Transfer outcome. This outcome occurs if
    - the bit transfer attempt is associated with a pair of corresponding source and destination user information bits,
    - the corresponding source and destination user information bits have the same binary value, and
    - the bit transfer attempt is included in a Block Transfer attempt in which the End of Block Transfer reference event occurs on or before the Block Transfer Timeout.

- Incorrect Bit outcome. This outcome occurs if
    - the bit transfer attempt is associated with a pair of corresponding source and destination user information bits,
    - the corresponding source and destination user information bits have different binary values, and
    - the bit transfer attempt is included in a block transfer attempt in which the End of Block Transfer reference event occurs on or before the Block Transfer Timeout.
- Misdelivered Bit outcome.[1] This outcome occurs if
    - the bit transfer attempt is associated with an extra user information bit (at an unintended destination),
    - the destination user information bit associated with the bit transfer attempt corresponds to a bit input to the system by the source user for transfer to an unintended destination user, and
    - the bit transfer attempt is not included in a block transfer attempt whose outcome is Refused Block.
- Extra Bit outcome.[2] This outcome occurs if the bit transfer attempt
    - is associated with an extra destination user information bit, and
    - is included in a block transfer attempt whose outcome is not Refused Block.

---

[1]When Bit Misdelivery Probability is not evaluated (as in the NTIA implementation), the bit transfer attempt that would result in a Misdelivered Bit outcome results in an Extra Bit outcome.

[2]This definition of Extra Bit outcome assumes that Bit Misdelivery Probability is not evaluated. If it is, see the standard for an alternate definition.

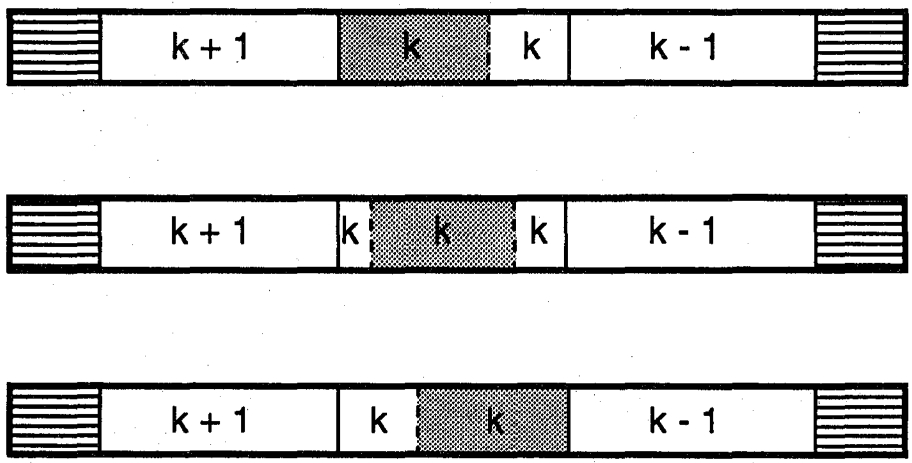
The rule for associating a string of extra destination user information bits with a certain block transfer attempt is rather involved. The rule is described in the following footnote and illustrated in Figure A-1.[3] The integer k refers to the k^{th} block transfer attempt with which the string is associated. The figure is divided into two, one for a block partially composed of Extra Bit outcomes (for which there are three possible scenarios) and one for a block completely composed of Extra Bit outcomes (for which there are four possible scenarios).

- <u>Lost Bit outcome</u>. This outcome occurs if the bit transfer attempt
  - is associated with an undelivered source user information bit, or
  - is associated with a pair of corresponding source and destination user information bits for which
    - End of Block Transfer reference event does not occur on or before Block Transfer Timeout occurs, and
    - the excessive delay is attributed to the system.
- <u>Refused Bit outcome</u>. The Refused Bit outcome occurs if it is contained in a block for which

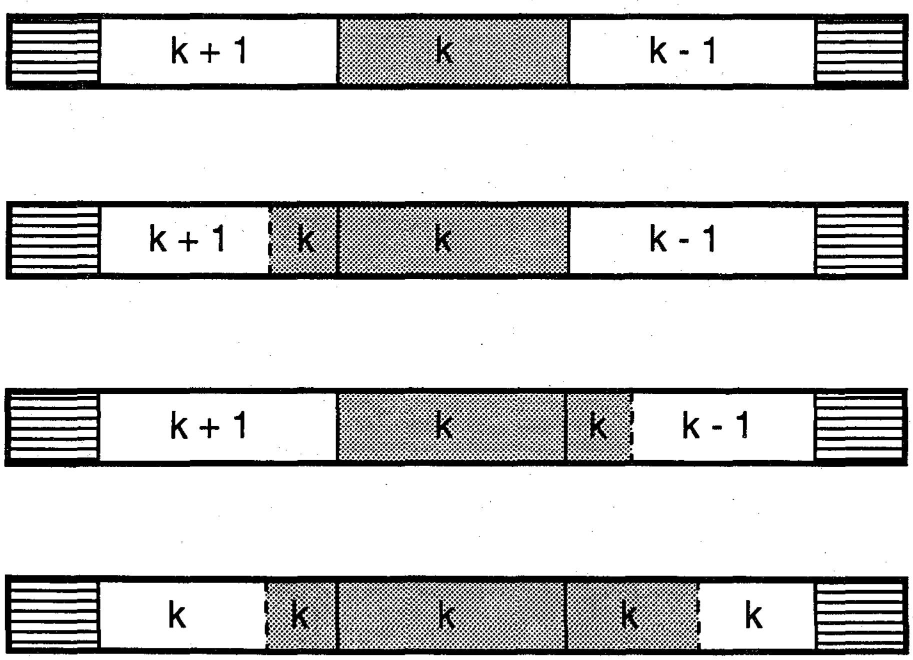
---

[3]A string of extra destination user information bits is shown as a shaded area. The string is associated with a block transfer attempt in the following way:

- <u>Partial Block of Extra Bits</u>. If the kth destination block contains both extra and nonextra bits, the extra bit string is associated with the same block transfer attempt as is the last nonextra bit preceding the string, unless the string is at the beginning of the block (in which case the extra string is associated with the same block transfer attempt as is the first nonextra bit following the string).
- <u>Complete Block of Extra Bits</u>. A destination block containing only extra bits is associated with a separate block transfer attempt unless the block is included in an extra bit string, both preceding and following nonextra bits associated with a common block transfer attempt (in which case, all bits in the destination block are associated with the same block transfer attempt as the bounding nonextra bits).



a. Partial Block of Extra Bits



b. Complete Block of Extra Bits

Figure A-1. Block transfer attempt containing Extra Bit outcomes.

- Start of Block Transfer reference event occurs,[4]

- End of Block Transfer reference event does not occur on or before Block Transfer Timeout, and

- the excessive delay is attributed to the user.[5]

Table A-1 summarizes the bit transfer outcomes.

A.2. Block Transfer Outcomes

The six possible user information block transfer outcomes are:

- Successful Block Transfer outcome. This outcome occurs if

    - Start of Block Transfer reference event occurs,

    - End of Block Transfer reference event occurs on or before Block Transfer Timeout, and

    - the destination user information bits associated with the block transfer attempt are identical to all of the source user information bits.

- Incorrect Block outcome. This outcome occurs if

    - Start of Block Transfer reference event occurs,

    - End of Block Transfer reference event occurs on or before Block Transfer Timeout, and

    - some (but not all) of the destination user information bits associated with the block transfer attempt are not identical to the source user information bits because

        - some (but not all) bits in the source block are undelivered user information bits,

        - some (but not all) bits in the destination block are extra user information bits, or

---

[4]Start of Block Transfer means that the system is authorized to transmit a single source user information block.

[5]These three conditions define the Refused Block outcome. It is also defined later with block transfer outcomes.

Table A-1. Bit Transfer Outcomes as a Function of Timeliness-Entity Responsibility, Accuracy, and Transfer.

				SOURCE BIT TRANSFERRED?	
				Yes	No
END OF BLOCK TRANSFER OCCURRED?	Yes	Destination Bit Received?	Yes	Successful Bit Transfer (Same Binary Value)	Extra Bit
				Incorrect Bit (Different Binary Value)	-
			No	Lost Bit (System Responsible)	-
	No			Lost Bit (System Responsible)	-
				Refused Bit (User Responsible)	-

- some (but not all) bits in the destination block are incorrect bits.

- Misdelivered Block outcome.[6] This outcome occurs if

  - the block transfer attempt is associated with a destination block consisting of extra user information bits, and

  - one or more destination user information bits associated with the block transfer attempt correspond to user information bits input to the system by the source user for transfer to some other destination user.

- Extra Block outcome.[7] This outcome occurs if the block transfer attempt is associated with a destination user information block consisting only of Extra Bit outcomes (i.e., Start of Block Transfer reference event does not occur).

- Lost Block outcome. This outcome occurs if

  - Start of Block Transfer reference event occurs,

  - End of Block Transfer reference event does not occur on or before Block Transfer Timeout, and

  - the excessive delay is attributed to the system.

- Refused Block outcome. The Refused Block outcome occurs if

  - Start of Block Transfer reference event occurs,

  - End of Block Transfer reference event does not occur on or before Block Transfer Timeout, and

  - the excessive delay is attributed to the user.

Table A-2 summarizes block transfer outcomes.

---

[6]When Block Misdelivery Probability is not evaluated, the bit transfer attempt that would result in a Misdelivered Block outcome is an Extra Block outcome.

[7]This definition assumes that Block Misdelivery Probability is not calculated. If it is, see the standard.

Table A-2. Block Transfer Outcomes as a Function of Timeliness-Entity Responsibility, Accuracy, and Start of Block Transfer

		START OF BLOCK TRANSFER OCCURRED?	
		Yes	No
END OF BLOCK TRANSFER OCCURRED?	Yes	Successful Block (Same Binary Value)	Extra Block
		Incorrect Block (Different Binary Value)	-
	No	Lost Block (System Responsible)	-
		Refused Block (User Responsible)	-

INDEX

2p 187
3c 187
3x 187
4r 187
5a 153, 187
6d 182, 187
Acceptance iv, xii, 1, 50, 57, 91, 93, 94, 96, 99, 152, 164, 165, 169, 170
Acceptance test xii, 93, 94, 164
ACCESS x, xi, xii, 1, 7, 9, 11, 15-22, 27, 28, 32, 34-36, 44, 57, 64, 66, 68, 79, 80, 90, 96, 97, 100, 103, 104, 111-114, 116, 122-124, 127, 143, 144, 153, 157, 160, 161, 167, 170, 172, 178
Access Confirmation 20-22, 32, 34, 35, 68, 104, 123
Access Denial 34, 36, 97, 124
Access Outage 34, 36, 124
Access Request 16, 19-21, 32, 35, 64, 104, 123
Access State 113
Access Time xii, 7, 35, 68, 97, 112, 124, 143, 157, 160, 161, 167, 170, 172
Access Timeout 32, 34, 66
Accuracy 29, 30, 38, 39, 43, 55, 59, 61, 68, 69, 97, 200
Achieved precision 162
ACK 15
ACO xi, 124, 127, 143, 144, 152
Addresses 15
Advanced Research Projects Agency Network (ARPANET) iii
all option 185
alyz 185
American National Standards Institute (ANSI) iii, 1
Analysis of variance (ANOVA) 1, 84
ANALYZ 121, 123, 127, 137-139, 143, 182
ANCILL 127, 137, 143
Ancillary performance parameter 35, 43, 49, 53, 131, 139, 157
Ancillary reference event 30, 62, 64
Application program 11-15, 50, 100, 103, 105, 108-112
ASCII 14, 15, 24, 100, 121, 122, 153, 181, 187
Autocorrelation 89-91, 152, 155, 160, 161, 175
Autocorrelation Coefficient 89, 160
Balance 89
Balanced sampling 60
Bias 89, 152
bin 184, 185
Binary 15, 37, 39, 43, 66, 69, 100, 153, 187, 193, 194, 200
Bit Comparison Outcomes x, 69, 128, 130
Bit Error Probability 39, 46, 135, 137
Bit Loss Probability 40, 46, 135
Bit Misdelivery Probability 37-40, 86, 137, 194
BITCOR 127, 133
Block Error Probability 7, 44, 137
Block Loss Probability 45
Block Misdelivery Probability 42, 45, 86, 137, 199

Block size 57, 90, 118, 120, 178
Block Transfer Time 43, 44, 69, 135, 137, 172, 175, 178
Block Transfer Timeout 36-38, 40, 42, 68, 69, 73, 128, 193-195, 197, 199
Blocking 20, 21, 27, 28, 30, 34, 51, 53, 59, 68, 89, 98, 103, 104, 108, 111, 112, 123, 124
Blocking design 98
box xii, 1, 30, 162, 172, 173, 175, 178, 182
Box plot 173
Calibration 114, 115
Call progress signal 16
CCITT 15, 16
Characteristics 58, 85, 96
Characterization iv, 1, 57, 90, 94-99, 187
Chronological plot 175
Circuit-switching 17
clean 184
code iv, xii, 2, 5-7, 14, 15, 103, 112, 113, 116, 118, 124, 135, 140, 143, 153, 180, 181, 183, 185, 187
COI 123
Combination of levels 57, 59, 81, 83, 86, 90, 91, 93-95, 155, 163, 167, 169
Command 112, 113, 115, 183, 184
Communication state 122, 143, 144
Comparison test 93
Compiler 183
Complete block 7, 195
Complete randomization 93
Completely randomized design 168
Confidence interval 81, 83, 89, 91, 152, 155, 157, 165
Confidence level 7, 57, 60, 81, 84, 89, 90, 165
Connection-oriented 17, 20, 21, 32, 35, 51, 79, 80, 103, 105, 111, 113
Connectionless 17, 32, 35, 79, 80
Control chart 94, 164
Conversion 18, 115, 181, 185
COR 127, 133
Correct Bit 130
Correlate 127
Correlation 38, 39, 43, 69, 144
Correlator 127, 133, 144
CSP 123
D10 xii, 140, 143, 152
d2 185
D20 xii, 140, 143, 152
d3a 185
d3b 185
d4 185
d5 153, 187
d6 187
Data analysis iv, 1, 6, 55, 81, 86, 88, 143, 152, 157, 181, 182, 187
Data communication function 11, 28-30, 55, 58, 64, 79, 96, 97, 144
Data communication process 9, 11, 28, 100, 108, 112
Data communication service iii, 5, 7, 59, 81, 94

Data communication session 1, 9, 11, 12, 16-21, 27, 28, 58, 66, 105, 108, 111, 131, 133
Data communication system xi, 5, 9, 11-13, 18, 55, 62, 100, 109, 112
Data display 1, 6, 86, 88, 181, 182, 187
Data extraction iv, xi, 1, 6, 7, 18, 55, 59, 61, 86, 100, 115, 116, 121, 153, 181, 185
Data medium 12, 13
Data Reduction iv, 1, 6, 55, 66, 86, 116, 121-123, 152, 181, 182, 185
Data terminal 13, 62
Datagram 17
Day of Week 157, 170
DCE 13, 16, 62
Degrees of freedom 160, 161
delay 29, 30, 34, 38, 42, 50, 51, 53, 64, 68, 80, 90, 97, 115, 153, 157, 161, 162, 170, 172, 173, 187, 195, 197, 199
Density 173, 183
Dependability 1, 26, 29, 30
Dependence 1, 60, 81, 89, 91, 98, 152, 155, 168, 175, 178
Dependent 9, 16, 89, 168, 175, 181, 191
Design iv, x, xi, xii, 1, 2, 5-7, 9, 11, 28, 30, 55, 57, 61, 77, 86, 88, 89, 90, 91, 93-98, 100, 103, 121, 122, 143, 152, 155, 168, 169, 181, 185, 191, 192
Destination Disengagement xii, 16, 27, 28, 79, 104, 112, 140, 143, 149, 172
Destination Disengagement Confirmation 16, 104
Destination Disengagement Denial 143
Destination Disengagement Request 104
Destination Disengagement Time 112, 143, 172
Destination Site 115, 157, 170
Destination user 14, 15, 17, 20-22, 24, 25, 36, 38, 42, 66, 69, 80, 97, 103, 104, 110, 116, 118, 120, 122, 127, 128, 144, 193-195, 197, 199
Destination user information bits 15, 36, 42, 97, 193-195, 197, 199
dev 183
Directory xii, 153, 180, 183-185, 187
DISENG 140
Disengagement Confirmation 16, 20, 27, 28, 51, 53, 54, 104, 140
Disengagement Denial 51, 54, 77, 140, 143
Disengagement Request 20, 27, 53, 54, 79, 104, 140
Disengagement Time xii, 53, 54, 112, 140, 143, 172, 173
Disengagement Timeout 51, 53, 73
Diskette 181, 182, 184, 185
Dispersion 90
Distribution 81, 98, 160, 161, 169
DLE 15
do shell script 116, 152, 153
doc 185
DOI xi, 116, 118, 122
DOS 180, 183
doscp 183, 184
DTE 13, 16, 19, 62
DUI xi, 116, 118, 122

Duplex 18, 29
elog 187
End of bit transfer 25
End of Block Transfer 20, 22, 24-26, 37, 38, 40, 42, 43, 62, 69, 104, 120, 128, 193-195, 197, 199, 200
End of user information transfer 114
End user 9, 11-14, 16, 21, 27-29, 34, 38, 51, 58, 60, 64, 68, 69, 77, 100, 103, 105, 108-112, 114, 115, 122, 124, 143
ENQ 15
Entity 11, 27, 38, 39, 43, 62, 64-66, 68, 69, 73, 77, 80, 103, 105, 112, 113, 128, 200
Entity-interface 64, 65, 113
EPILOG 121, 144, 182
Estimate 57, 77, 81, 83, 89-91, 94, 98, 99, 116, 152, 155, 160, 162, 163, 165, 167, 173, 175
Expected response 112
Experiment design iv, 1, 6, 30, 55, 57, 61, 89-91, 97, 121, 122, 152, 169, 181, 185
Extra Bit xiii, 37, 39, 46, 128, 130, 133, 135, 137, 193-195, 199
Extra Block 40, 42-44, 135, 137, 193, 199, 200
F statistic 160, 161
F77 183
Factor 29, 55, 57, 84, 85, 90, 93, 94, 96, 97, 152, 157, 163, 167-170
Factorial 59, 61, 98
fail 161
failac 153, 187
failb1 153
failb2 153
failb3 153
failb4 153
faild1 153
faild2 153
Failure xii, 21, 29, 34, 35, 39, 43, 46, 53, 58, 90, 97, 133, 135, 140, 143, 153, 162, 167, 172, 187
Fixed condition 97
Flag 14
Flow control 25, 62, 115
Flowchart x, xi, xii, 5, 66, 69, 73, 81, 124, 128, 140, 157
FORTRAN 99, 121, 122, 153, 180, 181, 183, 185, 187
Four-way handshake 54
Frame check sequence 15
Functional interface 12, 13, 62, 110
Gateway 62
Goodness of fit 173
grep 161
H0 167-169
Handshake 54
Histogram xii, 173
histogrm 182
History 77, 79, 80, 116, 118, 120, 124, 135, 140, 143
HP7475A 172, 178, 181

hpgraph 182
Hypothesis 1, 57, 81, 83, 84, 93-95, 155, 157, 161, 163-165, 167-170
Hypothesis test 81, 83, 84, 93, 163, 164, 167, 169
Imprecision 89
Incorrect Access 32, 35, 124
Incorrect Bit 37, 39, 40, 130, 133, 194
Incorrect Block 40, 43-45, 128, 133, 135, 193, 197, 200
Independent iii, 1, 2, 6, 8, 9, 16-18, 36, 61, 69, 79, 80, 89, 95, 111, 168, 175
Independent disengagement 79
Input Time iv, 25, 49, 50, 112, 144
Input/Output Time 25, 26, 45, 47, 49-51, 73, 79, 96, 128, 131, 135, 137, 138, 139, 162, 187
install 183, 184
Interaccess Delay 157, 170
Interblock gap 25
Interface x, 1, 9, 12-16, 18, 19, 21, 24, 28, 29, 32, 61, 62, 64-66, 73, 77, 80, 100, 103, 108-113, 116, 118, 120, 122, 183
Interface event 13, 16, 18, 62, 64, 65, 77, 111, 112
Interface monitor 18, 61, 66
Internetwork 62
Interval 57, 81, 83, 89, 91, 93, 99, 144, 152, 155, 157, 164, 165
Intra-system coordination information 15, 111
ISDN 15
Lag 90, 155, 160, 161
LAPB 15
Least squares 168, 175
Legend 178
Level 7, 57, 60, 81, 83, 84, 89, 90, 93, 95, 97, 99, 157, 161, 163, 165, 167, 169, 170
lib 187
Local interface 64, 65, 77, 80, 113, 116, 118
Local operating system 12
Local responsibility 144
Local user 18, 64, 65, 77, 80, 100
log.acc 161
log.wrk 161, 170
log.xfr 161
Lost Bit 38-40, 128, 130, 133, 137, 193, 195
Lost Block 42-45, 128, 130, 135, 137, 199, 200
Magnetic media 14
Maintenance iv, 91, 94, 96, 164
Management iv, 95, 96
Markov chain 81, 89, 152
Mathematical model 61
Mean 61, 93-95, 110, 160, 162, 164, 169
Measurement error 168
Menu file 178
merge 181
Microport System V.3 180, 183
Misdelivered Bit 37-40, 73, 130, 193, 194
Misdelivered Block 42, 43, 45, 73, 193, 199

Rootdir 184
runx 115
Sample size 1, 59, 60, 84, 98, 99, 155, 161, 165, 181, 191
Sampling xii, 60, 81, 93
Sampling error 81, 93
Satellite clock receiver 114, 115
Selection iv, 60, 95, 96, 98, 169, 170
Session profile xi, 1, 18, 58, 66, 103, 105, 108, 111-113
Shell script xii, 115, 116, 152, 153, 155, 161, 170, 185
Signal 16, 20, 21, 27, 28, 34, 51, 104, 123, 124
Significance level 57, 60, 83, 84, 161, 165, 167
Significance test 164
Simplex 18
Site 112, 115, 157, 160, 161, 170, 175, 178
Slope 168, 178
SOI xi, 116, 122
Source code 180, 183, 185, 187
Source Disengagement xii, 16, 27, 79, 104, 112, 140, 143, 149, 172, 173
Source Disengagement Confirmation 16, 104
Source Disengagement Denial 140
Source Disengagement Request 104
Source Disengagement Time xii, 112, 140, 143, 172, 173
Source Site 157, 160, 161, 170
Source user 14, 15, 20-22, 24, 25, 38, 42, 66, 69, 80, 100, 103, 104, 112, 113, 116, 118, 122, 193-195, 197, 199
Source user information bits 15, 197
Specification 123
Specifications x, xi, 1, 2, 5-8, 86, 122, 123, 144, 149, 172, 185
Specified performance value 46, 73
Specified precision 60, 81, 91, 98, 99, 152, 162
Speed 1, 29, 30, 97, 112, 178
SPI 122-124, 135, 143, 144, 185
spi.acd 122
spi.xfr 122
src 153, 185
star 99, 153, 155, 162, 167, 169, 170, 181, 182, 187
Start of bit transfer 25
Start of Block Transfer 20, 22, 24, 25, 40, 42, 43, 68, 69, 104, 118, 123, 193, 197, 199, 200
State 57, 77, 103, 105, 108, 112, 113, 116, 118, 122, 128, 143, 144
State code 112, 113, 116, 118
Structured design x, xi, xii, 1, 2, 5, 7, 9, 11, 28, 55, 77, 86, 88, 91, 100, 155
Structured design diagram x, xi, xii, 2, 5, 9, 11, 28, 55, 77, 86, 91, 100, 155
Subdirectory xiii, 185, 187
Subsystem 12, 13, 61, 62
Successful Access 32, 34, 35, 68, 97
Successful Bit Transfer 36, 39, 40, 49, 128, 133, 137, 138, 193
Successful Block Transfer 40, 43-45, 133, 197
Successful Disengagement 51, 53, 54, 77
SUI xi, 116, 118, 122

SUM 29, 34, 53, 79, 86, 123, 144, 168, 175
Superuser 184
Supported parameter 49
Switches 11, 178
SYN 15
System Blocking Signal 20, 21, 34, 104, 123
System control information 15, 111
System-Independent iii, 1, 6, 8, 9, 18, 61
Telephone system 17
Three-stage sampling 60
Threshold 46, 47, 93, 97, 99, 164, 165
Throughput xi, 7, 26, 47, 49, 95, 127, 131, 133, 137, 138, 144
Throughput trial 26, 127, 131, 133, 137
thrput 153, 187
Time of Day 157, 170
Timeout 29, 32, 34, 36-38, 40, 42, 51, 53, 66, 68, 69, 73, 79, 97, 115, 128, 193-195, 197, 199
tmp 185, 187
TRANSF 127
Transfer availability trial x, xi, 26, 45-48, 73, 97, 127, 128, 131, 133, 135
Transfer Denial Probability 47, 51, 97, 128, 139
Transfer sample xi, 22, 25, 26, 36, 45-47, 49-51, 68, 73, 79, 80, 127, 128, 131, 133, 135, 137-139
Transferred information 9, 14, 111
Transmission lines 11
Transport layer 62
Trial x, xi, 26, 45-48, 55, 59, 66, 73, 80, 90, 96, 97, 127, 128, 131, 133, 135, 137, 140, 143, 155, 162, 168, 185
TTSAMP 131, 133
Type I error 83, 99, 165
Type II error 84, 99, 165
UITIM 144
Unbalanced testing 59
Unbiased estimate 89
Undelivered Bit 130
Uniform iii, 1, 6, 8, 55
UNIXtm 115, 152, 161, 183
User blocking 20, 21, 30, 34, 68, 104, 123, 124
User Blocking Signal 20, 21, 34, 104, 123, 124
User control information 16, 111
User disengagement blocking 20, 27, 28, 51, 53, 104
User Disengagement Blocking Signal 20, 27, 28, 51, 104
User Fraction of Access Time 35, 68, 97, 124, 157, 160, 161
User Fraction of Block Transfer Time 44, 69, 135, 137
User Fraction of Disengagement Time 54
User Fraction of Input/Output Time 47, 49, 50, 96, 131, 135, 139, 162, 187
User Information Bit Transfer Rate 46, 47, 49, 51, 95, 96, 112, 131, 135, 138, 139, 162, 172, 187

User information transfer xii, 1, 5, 9, 11, 16, 18, 20-22, 24, 25, 28, 29, 36, 45, 58, 68, 96, 97, 103, 104, 111, 114, 122, 123, 127, 133, 137, 144, 161
User-Oriented iii, 1, 6, 8, 9, 121, 191, 192
User-Responsible Access Time 35
User-Responsible Block Transfer Time 44
User-Responsible Disengagement Time 54
User-Responsible Input Time 50
User-Responsible Input/Output Time 50
User-Responsible Output Time 50
User-system interface 12-15, 24, 77, 103, 108-113, 118, 120, 122
usr/data 153, 185, 187
usr/net xiii, 153, 184, 185, 187
Utilization 175
Variable condition 90, 93-95, 98, 152, 155, 157, 160, 161, 163, 164, 167, 168, 169, 170, 178
Venn diagram x, 62, 69
Virtual circuit 17
vp 182
vpaxes 178, 182
vplegend 182
X.25 15, 19, 64, 65
X.96 16
xmit_net 12, 100, 103, 112, 114

FORM **NTIA-29**
(4-80)

U.S. DEPARTMENT OF COMMERCE
NAT'L. TELECOMMUNICATIONS AND INFORMATION ADMINISTRATION

BIBLIOGRAPHIC DATA SHEET

	1. PUBLICATION NO.	2. Gov't Accession No.	3. Recipient's Accession No.
	95-319 (1)		

4. TITLE AND SUBTITLE Performance Evaluation of Data Communication Services: NTIA Implementation of American National Standard X3.141, Volume 1. Overview	5. Publication Date August 1995 6. Performing Organization Code NTIA/ITS.N3
7. AUTHOR(S) Martin J. Miles	9. Project/Task/Work Unit No.
8. PERFORMING ORGANIZATION NAME AND ADDRESS National Telecommunications and Information Admin. Institute for Telecommunication Sciences 325 Broadway Boulder, CO 80303	10. Contract/Grant No.
11. Sponsoring Organization Name and Address National Telecommunications and Information Admin. Herbert C. Hoover Building 14th and Constitution Avenue, NW Washington, DC 20230	12. Type of Report and Period Covered 13.

14. SUPPLEMENTARY NOTES

15. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

This volume is an overview of the American National Standards Institute (ANSI) specifications for performance evaluation of data communication services and the NTIA implementation. The specifications are described in ANS X3.102 (the standard that specifies the user-oriented and system-independent performance parameters) and ANS X3.141 (the standard that specifies uniform methods to evaluate them). The performance parameters are evaluated in four phases: experiment design, data extraction, data reduction, and data analysis. A fifth phase, data display, is an NTIA enhancement.

16. Key Words *(Alphabetical order, separated by semicolons)*

acceptance tests; access; American National Standards; analysis; analysis of variance (ANOVA); availability; box plots; characterization; data communication functions; data communication services; data communication session; dependability; dependence; disengagement; estimation; experiment design; extraction; factors; graphics; histograms; hypothesis tests; interface events; linear regression; performance parameters; precision; protocol; reduction; reference events; sample size; session profile; speed; structured design; user information transfer

17. AVAILABILITY STATEMENT	18. Security Class. *(This report)*	20. Number of pages
☒ UNLIMITED.	Unclassified	224
☐ FOR OFFICIAL DISTRIBUTION.	19. Security Class. *(This page)* Unclassified	21. Price:

*U.S. GOVERNMENT PRINTING OFFICE: 1995-676-490/25108