NTIA Report 95-319 (4)

Performance Evaluation of Data Communication Services: NTIA Implementation of American National Standard X3.141 Volume 4. Data Reduction

Kenneth P. Spies

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

Larry Irving, Assistant Secretary
for Communications and Information

August 1995

CONTENTS

Page

FIGURES iv

ABSTRACT 1

1. INTRODUCTION 1

2. SYNOPSIS OF DATA REDUCTION 3

3. DATA REDUCTION INPUT 7
   3.1 Reduction Specifications 7
   3.2 Performance Data 12
      3.2.1 Communication State Model 13
      3.2.2 Overhead Information Files 18
      3.2.3 User Information Files 23
   3.3 Examination of Input Data 29
      3.3.1 Examination of Specifications Input File 30
      3.3.2 Examination of Performance Data Files 30
   3.4 Consolidation of Reference Event Data 33

4. PERFORMANCE ASSESSMENT 38
   4.1 Access Performance Assessment 38
      4.1.1 Identification of Access Attempts 38
      4.1.2 Determination of Access Outcomes 40
   4.2 User Information Transfer Performance Assessment 46
      4.2.1 Identification of Bit and Block Transfer Attempts: Data Correlation 47
      4.2.2 Determination of Bit and Block Transfer Outcomes 99
      4.2.3 Availability and Throughput Transfer Samples 109
   4.3 Disengagement Performance Assessment 119
      4.3.1 Identification of Disengagement Attempts 121
      4.3.2 Determination of Disengagement Outcomes 124
   4.4 Performance Time Allocation 127

5. PRODUCTION OF PERFORMANCE ASSESSMENT SUMMARY 134

6. ACKNOWLEDGEMENTS 144

7. REFERENCES 145

APPENDIX: SHELL SCRIPT IMPLEMENTATION OF REDUCTION PROGRAMS 147

FIGURES

Page

Figure 1. Outline of data reduction scheme. 4

Figure 2. Record sequence in specifications input file. 8

Figure 3. Record formats in specifications input file. 9

Figure 4. Examples of specifications input files. 10

Figure 5. Summary of communication state model. 14

Figure 6. Record sequence in overhead information files. 19

Figure 7. Record formats in overhead information files. 20

Figure 8. Examples of overhead information files. 21

Figure 9. ASCII-character representation of user information. 24

Figure 10. Record sequence in user information files. 25

Figure 11. Record formats in user information files. 26

Figure 12. Examples of user information files. 27

Figure 13. Record sequence and record formats in consolidated overhead information file. 34

Figure 14. Event records in a typical consolidated event history. 35

Figure 15. Determination of communication states in the consolidated event history. 37

Figure 16. Scheme for determining access outcomes. 42

Figure 17. Record sequence and record formats in access outcome file. . . 44

Figure 18. Example of an access outcome file. 45

Figure 19. Outline of user information transfer phase of normal data communication session. 48

Figure 20. Outline of data correlation process. 50

Figure 21. Organization of Section 4.2.1. 51

Figure 22. User information bit categories. 52

Figure 23. Bit comparison outcome (BCO) categories. 54

FIGURES (Cont'd)

Page

Figure 24. Out-of-sequence bits. 55

Figure 25. Configurations of corresponding bit pairs and undelivered bits associated with a pair of source and destination bit sequences. 57

Figure 26. Patterns of identical source and destination bit strings produced by isolated bit transfer failures. 60

Figure 27. Summary of algorithm E for identifying clusters of incorrect BCOs. 62

Figure 28. Summary of algorithm U for identifying strings of undelivered BCOs. 65

Figure 29. Typical multi-sequence set $\hat{Q}$ of most probable BCO sequences. 67

Figure 30. Summary of algorithm X for identifying strings of extra BCOs. 68

Figure 31. Outline of overall BCO identification process used by BITCOR. 70

Figure 32. Effect of user information window size on BITCOR correlation performance. 75

Figure 33. User information window size and BITCOR correlation performance. 78

Figure 34. Scheme for finding appropriate user information window size. 80

Figure 35. Example of poorly-isolated bit transfer failures. 81

Figure 36. Reconfiguration of user information block boundaries. 83

Figure 37. Block transfer attempts covering reconfigured user information block boundaries. 86

Figure 38. Block transfer attempts containing undelivered BCOs. 88

Figure 39. Block transfer attempts containing both extra and nonextra BCOs. 89

Figure 40. Block transfer attempts containing only extra BCOs. 90

Figure 41. Outline of BCO assignment algorithm. 92

FIGURES (Cont'd)

Page

Figure 42. Examples of block transfer event time determination. 94

Figure 43. Record sequence in correlator output file. 96

Figure 44. Record formats in correlator output file. 97

Figure 45. Example of a correlator output file. 98

Figure 46. Scheme for determining block transfer and bit transfer outcomes. 102

Figure 47. Record sequence and record formats in block transfer outcome file. 105

Figure 48. Example of a block transfer outcome file. 106

Figure 49. Bit transfer outcome file. 108

Figure 50. Effect of mislocated undelivered BCOs on block transfer outcomes. 110

Figure 51. Summary of transfer sample concepts. 112

Figure 52. Scheme for determining availability transfer sample outcomes. 116

Figure 53. Transfer sample outcome file. 118

Figure 54. Throughput sample outcome file 120

Figure 55. Scheme for determining disengagement outcomes. 126

Figure 56. Record sequence and record formats in disengagement outcome files. 128

Figure 57. Examples of source and destination disengagement outcome files. 129

Figure 58. Outline of performance time allocation. 130

Figure 59. Local and overall responsibility states. 132

Figure 60. Organization of EPILOG processes. 135

Figure 61. Example of assessment summary preface. 136

Figure 62. Example of access assessment summary. 137

FIGURES (Cont'd)

Page

Figure 63 (Part 1). Example of user information transfer assessment summary. 139

Figure 63 (Part 2). Example of user information transfer assessment summary. 140

Figure 63 (Part 3). Example of user information transfer assessment summary. 141

Figure 64. Example of source disengagement assessment summary. 142

Figure 65. Example of destination disengagement assessment summary. . . . 143

Figure A-1. Structured design diagram of data reduction process. 148

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

VOLUME 4. DATA REDUCTION

K. P. Spies[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 shows how the data reduction phase is implemented by a set of FORTRAN computer programs and associated I/O files. These programs examine extracted performance data to identify individual trials and determine their outcomes. This volume outlines the data reduction process, describes input to the process, and discusses key concepts used by data reduction procedures.

Key words: American National Standards; data communications; end users; performance measurements; performance parameters

1. INTRODUCTION

In the NTIA implementation of ANS X3.141 (ANSI, 1987), the data reduction phase of data communication performance evaluation is accomplished by a set of FORTRAN computer programs and associated I/O files. This volume describes key features of the data reduction process and contains four principal sections. A synopsis of the data reduction process is presented in Section 2. Section 3 includes a detailed description of input to the reduction process; it defines a communication state model that underlies the representation of reference events in the extracted performance data files, specifies record formats for all input data files, and describes preliminary procedures that subject input data to a series of validity checks. Section 3 also describes a procedure that combines reference event data observed at the source and destination interfaces to produce a unified event history. Section 4 discusses concepts used by performance assessment procedures to identify performance trials and determine their

[1]The author is with the Institute for Telecommunication Sciences, National Telecommunications and Information Administration, U.S. Department of Commerce, Boulder, CO 80303-3328.

outcomes. Section 5 outlines the production of a summary of assessment results and shows a typical assessment summary for each primary function. An appendix describes the implementation of the reduction programs in a comprehensive shellscript that processes a single performance measurement test.

2. SYNOPSIS OF DATA REDUCTION

The principal task in the data reduction phase of data communication performance evaluation is the identification of individual performance trials (e.g., access attempts) and the determination of their outcomes. In the NTIA measurement system, this is accomplished by a set of FORTRAN computer programs and associated I/O files. The data reduction scheme is outlined in Figure 1. Extracted performance data are processed by a sequence of three main programs: **PROLOG**, **ANALYZ**, and **EPILOG**. Each execution of this sequence is called a reduction run.

Input files for a reduction run, shown at the left in the figure, contain extracted performance data and a set of reduction specifications. The performance data files for a reduction run, called a performance data batch, contain reference event records that describe the service provided by a given data communication system to a specified source and destination user pair during a selected observation period. A performance data batch consists of four files: a source overhead information file (SOI), a source user information file (SUI), a destination overhead information file (DOI), and a destination user information file (DUI). Each overhead information file contains records of all primary access and disengagement reference events and all significant ancillary reference events observed at the local user-system interface during the monitored sessions. Each user information file contains records of all user information blocks transferred across the local user-system interface during the monitored sessions. All performance data files consist of formatted (ASCII-character) records and must conform to the specifications presented in Section 3.2.

The specifications input file (**SPI**) for a reduction run contains an identifier for the performance data batch to be processed and a set of numerical data items used in reduction procedures. The latter includes specified parameter values for determining outcomes of performance trials in accordance with ANS X3.141. A detailed description of the specifications input file is given in Section 3.1.

Each main program executed in a reduction run implements a particular phase of the overall reduction process as outlined in Figure 1. In the first phase, implemented by **PROLOG**, preliminary procedures examine input data files for compliance with format and content requirements. If no errors are detected, consolidation procedures (i) combine reference event data in the source and

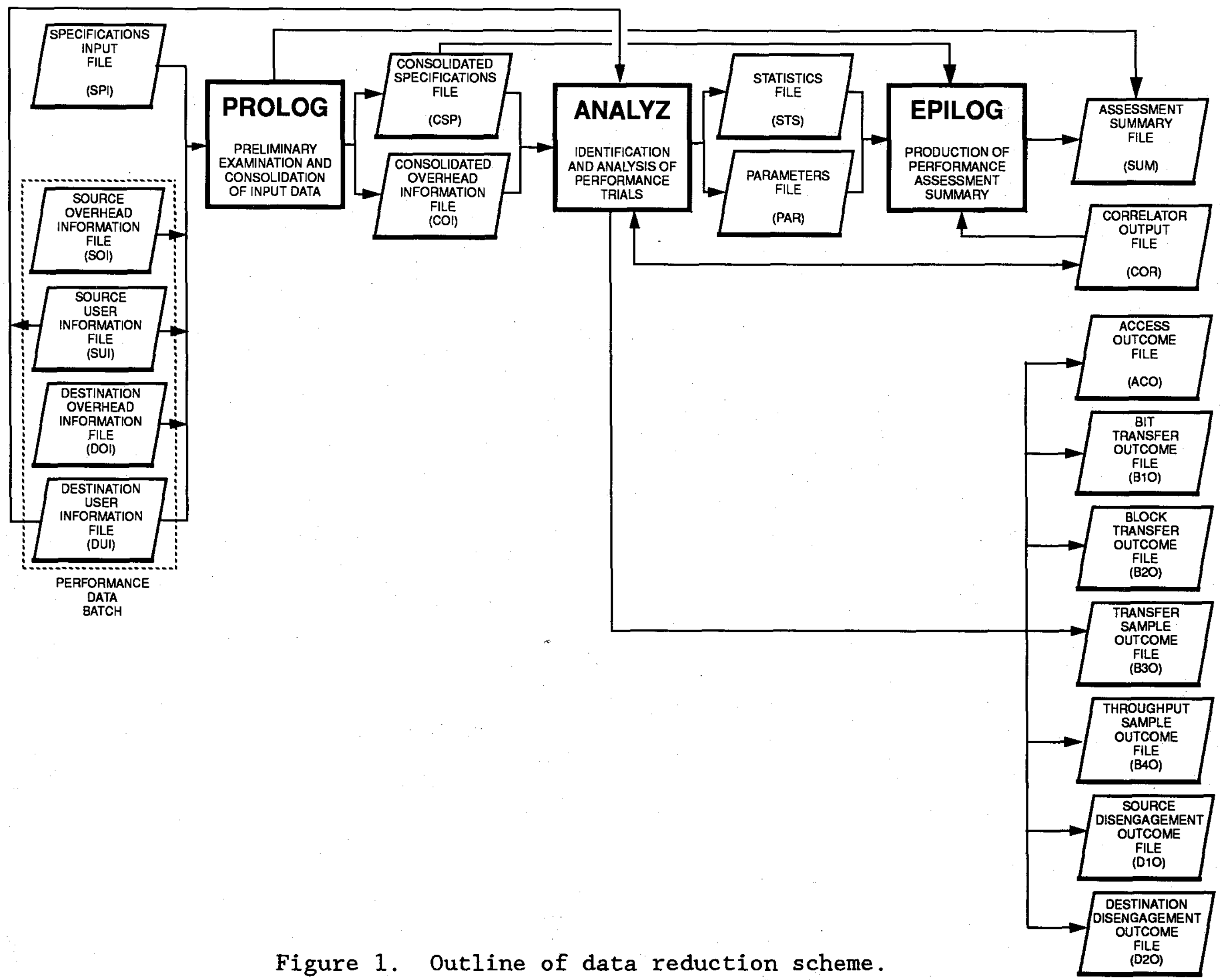


Figure 1. Outline of data reduction scheme.

destination overhead information files to produce a unified event history in the consolidated overhead information file (**COI**) and (ii) combine data read from the specifications input file and the overhead information files to produce a comprehensive set of reduction specifications in the consolidated specifications file (**CSP**).

If the preliminary examination of input data detects an error, a diagnostic is written to the assessment summary file (**SUM**); the reduction run is called a suspended run and further data processing in the run is suppressed. Otherwise, the run is called a normal reduction run. Preliminary examination procedures are described in Section 3.

In the second phase of a normal reduction run, program **ANALYZ** implements the identification and analysis of performance trials. Separate subroutines carry out assessment procedures for the access, user information transfer, and disengagement functions. Assessment procedures for these functions are described in Sections 4.1, 4.2, and 4.3, respectively. Each assessment routine examines reference event records in the consolidated overhead information file or the user information files to identify individual performance trials and determine their outcomes. As each outcome is determined, it is recorded in the appropriate performance outcome file (shown at the right in Figure 1). The outcome record for a successful trial contains both overall and user performance times, whereas the record for an unsuccessful trial specifies the particular failure outcome (e.g., Access Denial, Incorrect Block). When all trials associated with a function have been identified and analyzed, the relevant assessment routine calculates measured values of primary and ancillary ANS X3.102 (ANSI, 1983) performance parameters for the function. After assessment procedures have been completed for all functions, performance statistics (outcome counts and cumulative performance times) and measured parameter values are written to the statistics file (**STS**) and the parameters file (**PAR**), respectively. These files are used to pass assessment results to the **EPILOG** program.

In user information transfer performance assessment, special data correlation routines compare source (transmitted) and destination (received) user information to identify bit and block transfer attempts; these are recorded in the correlator output file (**COR**). The contents of that file are subsequently analyzed to determine bit and block transfer outcomes. Misdelivery performance is not evaluated.

Delay attributable to user activities during the performance of a function is required to estimate values of ancillary performance parameters and assign responsibility for timeout failures to the system or to users. User delay for an individual trial is obtained by invoking a performance time allocation routine. This routine examines the consolidated event history for the associated performance period and identifies intervals of overall user responsibility in accordance with ANS X3.141. Performance time allocation is described in Section 4.4.

In the final phase of a normal reduction run, program **EPILOG** implements the production of a user-oriented summary of assessment results. This summary, which is written to the assessment summary file (**SUM**), lists test descriptors, observed outcome counts, measured performance parameter values, and specified parameter values used in determining outcomes in accordance with ANS X3.141. An outline of production procedures and printouts of a typical assessment summary are given in Section 5.

3. DATA REDUCTION INPUT

This section presents a detailed description of the input data files for a reduction run. As outlined in the preceding section, these include (i) the specifications input file (SPI) containing reduction specifications for the run and (ii) a set of overhead and user information files constituting the performance data batch to be processed. The former is described in Section 3.1 and the latter is described in Section 3.2. Section 3.3 discusses procedures that examine input data files for compliance with format and content requirements. Section 3.4 describes a procedure for combining reference event data in the source and destination overhead information files to produce a unified event history.

3.1 Reduction Specifications

The specifications input file (SPI) for a reduction run contains an ASCII-character batch identifier and a set of numerical data items used in reduction procedures. The latter includes specified parameter values for determining performance trial outcomes in accordance with ANS X3.141. These specified values generally represent expected values derived from previous measurements or are based on user requirements.

The sequence of records in the specifications input file is shown in Figure 2. Note that the record sequence for a particular reduction run depends on the set of primary data communication functions - access, user information transfer, or disengagement - to be assessed in the run. Formats and contents of individual records are summarized in Figure 3; additional details are provided in the paragraphs that follow. Examples of specifications input files are illustrated in Figure 4. The specifications input file is referenced in every reduction run. It is written prior to starting the run and must conform to conditions presented in this section.

Preface Data. This record contains a file descriptor consisting of the character string SPECIFICATIONS INPUT left-justified in a 32-character field with blank fill on the right. During the preliminary examination of input data in **PROLOG**, the descriptor field read from the SPI file is compared with the prescribed (expected) descriptor. Any difference between the two fields is a fatal input data error and further processing in the reduction run is suppressed as described in Section 3.3.

PREFACE DATA
IDENTIFIERS
ASSESSMENT OPTIONS
ACCESS SPECIFICATIONS[1]
TRANSFER SPECIFICATIONS (PART 1)[2]
TRANSER SPECIFICATIONS (PART 2)[2]
CORRELATOR SPECIFICATIONS[2]
DISENGAGEMENT SPECIFICATIONS[3]

[1] Record is included when access or disengagement performance assessment is enabled.

[2] Record is included when user information transfer performance assessment is enabled.

[3] Record is included when disengagement performance assessment is enabled.

Figure 2. Record sequence in specifications input file.

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
PREFACE DATA:		
1-32	A32	FILE DESCRIPTOR
IDENTIFIER:		
1-64	A64	BATCH IDENTIFIER
ASSESSMENT OPTIONS:		
1-4	I4	ACCESS ASSESSMENT OPTION
5-8	I4	USER INFORMATION TRANSFER ASSESSMENT OPTION
9-12	I4	DISENGAGEMENT ASSESSMENT OPTION
ACCESS SPECIFICATIONS:		
1-16	E16.0	SPECIFIED VALUE OF ACCESS TIME (SECONDS)
17-32	E16.0	SPECIFIED VALUE OF USER FRACTION OF ACCESS TIME
TRANSFER SPECIFICATIONS (PART 1):		
1-16	E16.0	SPECIFIED VALUE OF BLOCK TRANSFER TIME (SECONDS)
17-32	E16.0	SPECIFIED VALUE OF USER FRACTION OF BLOCK TRANSFER TIME
33-48	E16.0	SPECIFIED VALUE OF USER INFORMATION BIT TRANSFER RATE FOR A TRANSFER AVAILABILITY TRIAL(BITS/SECOND)
49-64	E16.0	SPECIFIED VALUE OF USER FRACTION OF INPUT/OUTPUT TIME FOR A TRANSFER AVAILABILITY TRIAL

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
TRANSFER SPECIFICATIONS (PART 2):		
1-16	E16.0	SPECIFIED VALUE OF BIT ERROR PROBABILITY FOR A TRANSFER AVAILABILITY TRIAL
17-32	E16.0	SPECIFIED VALUE OF BIT LOSS PROBABILITY FOR A TRANSFER AVAILABILITY TRIAL
33-48	E16.0	SPECIFIED VALUE OF EXTRA BIT PROBABILITY FOR A TRANSFER AVAILABILITY TRIAL
49-56	F8.0	MINIMUM NUMBER OF BIT TRANSFER ATTEMPTS IN A TRANSFER AVAILABILITY TRIAL
CORRELATOR SPECFICATIONS:		
1-8	I8	USER INFORMATION WINDOW SIZE (BITS)
9-16	F8.0	MAXIMUM BIT SHIFT IN INCORRECT BIT IDENTIFICATION ALGORITHM
17-24	F8.0	MAXIMUM BIT SHIFT IN UNDELIVERED BIT IDENTIFICATION ALGORITHM
25-32	F8.0	MAXIMUM BIT SHIFT IN EXTRA BIT IDENTIFICATION ALGORTIHM
DISENGAGEMENT SPECIFICATIONS:		
1-16	F16.0	SPECIFIED VALUE OF SOURCE DISENGAGEMENT TIME (SECONDS)
17-32	F16.0	SPECIFIED VALUE OF USER FRACTION OF SOURCE DISENGAGEMENT TIME
33-48	F16.0	SPECIFIED VALUE OF DESTINATION DISENGAGEMENT TIME (SECONDS)
49-64	F16.0	SPECIFIED VALUE OF USER FRACTION OF DESTINATION DISENGAGEMENT TIME

Figure 3. Record formats in specifications input file.

```
SPECIFICATIONS INPUT
NTIA - PDN test from Washington, DC                                  0915
   1    0    1
          4.5E+01         5.00E-02
          1.4E+01         7.50E-02          4.0E+00          2.0E-01
```

a. File for Reduction of Access-Disengagement Test Data

```
SPECIFICATIONS INPUT
NTIA - ITS (Boulder)                                                 1424
   0    1    0
         3.00E+00          5.0E-01          1.0E+04          5.0E-01
          1.0E-08          1.0E-08          1.0E-08   30000.
       16      256.    8192.    8192.
```

b. File for Reduction of User Information Transfer Test Data

Figure 4. Examples of specifications input files.

Identifiers. This record consists of an ASCII-character batch identifier uniquely associated with the performance data batch to be processed in the reduction run. This identifier is contained in a 64-character field that must be identical to the batch identifier field in each overhead and user information file for the run (see Section 3.2). Any difference among these batch identifier fields is a fatal input data error. The NTIA implementation of ANS X3.141 includes the test number as characters 61-64 in the batch identifier.

Assessment Options. Values of the access, user information transfer, and disengagement assessment options specify whether performance assessment for the corresponding function is enabled (1) or suppressed (0). Performance assessment for access, user information transfer, and disengagement may be enabled in any combination that includes at least one function. A fatal input data error results if any option differs from both 0 and 1 or if all options are 0.

Access Specifications. This record contains specified values of Access Time and User Fraction of Access Time used to determine outcomes of access attempts in accordance with ANS X3.141. The access specifications record is included in the specifications input file only when access or disengagement performance assessment is enabled (the disengagement assessment routine uses the specified value of Access Time to identify successful access attempts).

Transfer Specifications (Part 1). This record contains specified values of Block Transfer Time and User Fraction of Block Transfer Time used to determine outcomes of bit and block transfer attempts in accordance with ANS X3.141. The record also contains specified values of User Information Bit Transfer Rate and User Fraction of Input/Output Time used to determine availability transfer sample outcomes in the measurement of Transfer Denial Probability.

Transfer Specifications (Part 2). This record contains specified values of the supported bit transfer failure probabilities - Bit Error Probability, Bit Loss Probability, and Extra Bit Probability - used to determine availability transfer sample outcomes in Transfer Denial measurements. This record also specifies the minimum number of bit transfer attempts to be included in an availability transfer sample. Guidelines for selecting the size of a transfer sample are discussed in Volume 2 (Section 5.2) of this report. Transfer specifications records and the subsequent correlator specifications record are included in the specifications input file only if user information transfer performance assessment is enabled in a reduction run.

Correlator Specifications. This record contains specifications used by algorithms for identifying clusters of incorrect bits, strings of undelivered bits, and strings of extra bits. These algorithms systematically select and compare strings of uncorrelated source and destination bits whose length is specified by the user information window size. The maximum bit shift in the incorrect bit identification algorithm specifies the longest cluster of incorrect bits that can be identified by the algorithm. Similarly, maximum bit shifts in the undelivered bit and extra bit identification algorithms specify the longest strings of undelivered and extra bits that can be identified by the respective algorithms. If the length of a cluster of incorrect bits, a string of undelivered bits, or a string of extra bits exceeds the relevant specification in this record, the correlation process cannot be completed. The contents of this record are defined and discussed in Section 4.2.1. As described there, these specifications are used in comparing source and destination user information to identify bit and block transfer attempts.

Disengagement Specifications. The NTIA implementation of ANS X3.141 segregates source and destination disengagement attempts in separate measurement samples and calculates separate estimates of source and destination disengagement parameters. The disengagement specifications record contains specified values of Disengagement Time and User Fraction of Disengagement Time used to determine outcomes of source and destination disengagement attempts. This record is included in the specifications input file only if disengagement performance assessment is enabled in a reduction run.

3.2 Performance Data

The set of performance data files for a reduction run, called a performance data batch, contains reference event records that describe, in accordance with ANS X3.141, the service provided by a given data communication system to a specified source and destination user pair during a selected observation period. A performance data batch consists of four files: a source overhead information file (SOI) and a source user information file (SUI) that contain performance data extracted at the source user-system interface, and a destination overhead information file (DOI) and a destination user information file (DUI) that contain performance data extracted at the destination user-system interface. All performance data files input to the NTIA reduction programs consist of formatted

(ASCII-character) records and must conform to the specifications presented in this section.

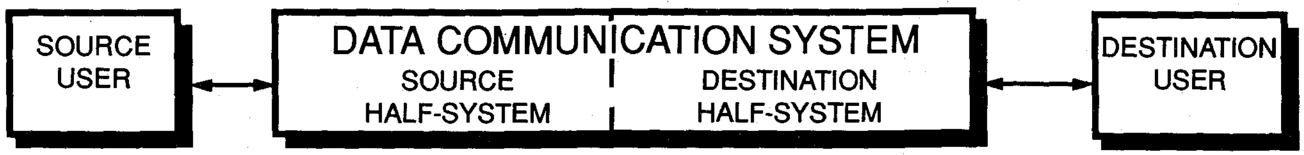
The observation period corresponding to a performance data batch may consist of a single data communication session, a succession of sessions separated by idle intervals, or a portion of a session. A single user must serve as the source user in all sessions in a given batch and another user must serve as the destination user in all sessions. A specified user, which may be either the source or destination user, serves as the originating user in all sessions in a batch. All sessions encompassed by a performance data batch must be in the same category; i.e., all sessions must be connection oriented or all must be connectionless. The initial disengagement attempt in each session in a batch must belong to the same category - negotiated or independent. In connectionless sessions, the source user is the originating user and the initial disengagement attempt is independent.

Each overhead information file contains records of all primary access and disengagement reference events and all significant ancillary reference events observed at the local user-system interface during the monitored sessions. The representation of these events is based on a state model of the underlying data communication process. Each user information file contains records of user information blocks transferred across the local user-system interface during the monitored sessions. Data recorded for each block include the relevant user information transfer event times and the binary content or representation of the transferred user information.

This section presents a detailed description of performance data batches and consists of three subsections. The first discusses a state model of the data communication process and the next two describe overhead and user information files, respectively.

3.2.1 Communication State Model

The communication state model used to represent reference events recorded in the overhead information files is summarized in Figure 5. The model includes four communicating entities: a source and destination pair of end users receiving service and a source and destination pair of "half-systems" providing service. Each half-system represents the portion of the end-to-end data communication system that interacts with the adjacent user. This division of the



a. Model Entities

PRIMARY COMMUNICATION STATE	ANCILLARY COMMUNICATION STATE	COMPOSITE COMMUNICATION STATE CODE
IDLE	WAITING	0
	ACTIVE	1
COMMITTED	WAITING	2
	ACTIVE	3
CLOSING	WAITING	4
	ACTIVE	5

b. Communication States

Figure 5. Summary of communication state model.

data communication system into two conceptually separate entities reflects the fact that different system activities may be simultaneously underway at the two user interfaces during a portion of a data communication session. Each entity is represented by a simple finite-state machine characterized, at any given time, by a specific communication state that describes the involvement of that entity in a given data communication session. Primary access and disengagement reference events and all ancillary reference events occurring in the session are then represented by discrete changes in the communication state of one or more model entities.

Relative to a given data communication session, each entity is in one of the three primary communication states defined below.

1. Idle State. The entity is not involved in the given session. (The entity may be involved in another session, or may not be involved in any session.)

2. Committed State. The entity is involved in the given session with the intent to transfer (transmit or receive) additional user information. (The entity is carrying out access or user information transfer activities.)

3. Closing State. The entity is involved in the given session with the intent to terminate its involvement without transferring additional user information. (The entity is carrying out disengagement activities.)

Each primary state includes two ancillary communication states: the active state and the waiting state. These have different meanings that depend on the associated primary state. Within the committed and closing states, the ancillary states describe an entity's responsibility for producing the next event at the local user-system interface. If an entity is responsible for producing the next event, the entity is in the active state; otherwise, the entity is in the waiting state. Within the idle state, the ancillary states describe an entity relative to designated or scheduled service time intervals during which the entity may participate in data communication activities. When an entity is within a service time interval, but is not involved in the given session, the entity is in the active state. When an entity is not within a service time interval, it is in the waiting state. A transition between the idle-active and idle-waiting states corresponds to the beginning or end of a service time interval.

Together, the three primary and the two ancillary states result in a total of six composite communication states. Each composite communication state is represented in the source and destination event histories by a numerical communication state code as shown in Figure 5b. Note that, at a particular interface, no more than one committed or closing entity can be in an active ancillary state. However, both entities at an interface may be simultaneously in waiting states.

Primary access and disengagement reference events recorded in overhead information files correspond to particular primary communication state transitions by one or more entities. Relationships between these reference events and the corresponding model events (communication state transitions) are specified in the following paragraphs.

Access Request. This event notifies the system of a user's desire for data communication service. It initiates a data communication session and begins the access function. In the communication state model, an Access Request is represented by an event in which the originating user and the adjacent half-system undergo transitions from the idle-active (1) state to a committed (2 or 3) state.

Nonoriginating User Commitment. This event expresses the intent of the nonoriginating user to participate in a requested data communication session. It is represented by a model event in which the nonoriginating user undergoes a transition from the idle-active (1) state to a committed (2 or 3) state. If the adjacent half-system has not already entered a committed state, that entity undergoes the same primary state transition.

System Blocking Signal. This event notifies the originating user that the system cannot provide service in a requested data communication session. The originating user and the adjacent (issuing) half-system undergo transitions from a committed (2 or 3) state to a closing (4 or 5) state. Because the same primary state transition may be associated with a User Blocking Signal or a Disengagement Request, a System Blocking Signal is represented by two successive model events having a common event time. In the first, the issuing half-system enters a closing state and the adjacent (originating) user remains in a committed state. In the second event, the originating user enters a closing state and the adjacent half-system remains in a closing state.

User Blocking Signal. This event, which is the user's counterpart to a System Blocking Signal, notifies the system that the issuing user will not participate in a requested data communication session. The issuing user and the adjacent half-system undergo transitions from a committed (2 or 3) state to a closing (4 or 5) state. To avoid the ambiguity noted previously, a User Blocking Signal is also represented by two successive model events having a common event time. In the first, the issuing user enters a closing state and the adjacent half-system remains in a committed state. In the second event, the adjacent half-system enters a closing state and the issuing user remains in a closing state.

Disengagement Request. This event requests termination of a user's participation in an established data communication session (a session in which user information transfer has occurred or may occur). A Disengagement Request is represented by a single model event in which the disengaging user and the adjacent half-system undergo transitions from a committed (2 or 3) state to a closing (4 or 5) state.

Disengagement Confirmation. This event verifies that the local (disengaging) user's participation in an established data communication session has been terminated. A Disengagement Confirmation is represented by a single model event in which the disengaging user and the adjacent half-system undergo transitions from a closing (4 or 5) state to an idle (0 or 1) state.

An ancillary reference event, as described in ANS X3.141, may affect responsibility states at the local interface, at the remote interface, or at both interfaces. The local effect is represented in the communication state model by appropriate transitions in the ancillary states of local entities. The remote effect of an ancillary event is described by an associated remote interface effect code. This code is 1 if the event conveys responsibility to the system for producing a subsequent event at the remote interface and is 0 otherwise. An interface event may correspond to both a primary reference event and an ancillary reference event. In such cases, both reference events may be represented by a single model event (except as described earlier for blocking signals) in which affected (local) entities undergo both primary and ancillary state transitions.

3.2.2 Overhead Information Files

As outlined previously, the source overhead information file (SOI) and the destination overhead information file (DOI) in a performance data batch contain chronologically ordered records of reference events observed at the respective user-system interfaces during a measurement period. These reference events are represented in the overhead information files by particular transitions in the communication state of one or more entities according to the data communication process model described in the preceding section.

The sequence of records in an overhead information file is shown in Figure 6. Formats and contents of individual records are summarized in Figure 7; additional details are provided in the paragraphs that follow. Examples of overhead information files are illustrated in Figure 8. Source and destination overhead information files are referenced in every reduction run and must conform to the specifications presented in this section.

Preface Data (Part 1). This record contains a file descriptor consisting of the character string SOURCE OVERHEAD INFORMATION in the source overhead information file and the string DESTINATION OVERHEAD INFORMATION in the destination overhead information file. Each descriptor is left-justified in a 32-character field with blank fill on the right. During the preliminary examination of input data in **PROLOG**, the descriptor field read from the **SOI** or **DOI** file is compared with the prescribed (expected) descriptor field. Any difference between the two fields is a fatal input data error and further processing in the reduction run is suppressed.

Preface Data (Part 2). This record consists of an ASCII-character batch identifier in a 64-character field. If the identifier consists of fewer than 64 characters, blank fill must be used to complete the field. The two overhead information files in a performance data batch must contain identical batch identifier fields.

Preface Data (Part 3). This record consists of ASCII-character source and destination user identifiers. Each identifier is contained in a 32-character field. If either identifier consists of fewer than 32 characters, blank fill must be used to complete the field. The two overhead information files in a performance data batch must contain identical source user identifier fields and identical destination user identifier fields.

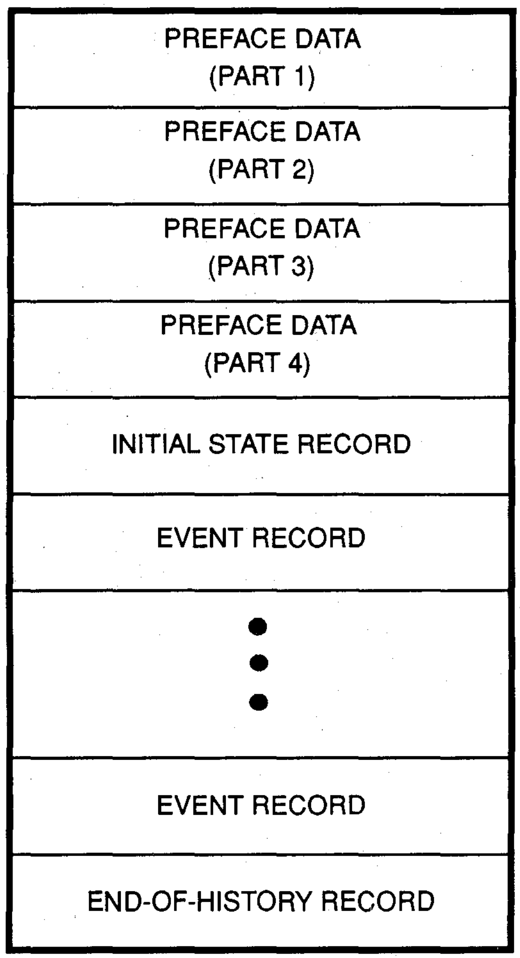


Figure 6. Record sequence in overhead information files.

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS	
PREFACE DATA (PART 1):			
1-32	A32	FILE DESCRIPTOR	
PREFACE DATA (PART 2):			
1-64	A64	BATCH IDENTIFIER	
PREFACE DATA (PART 3):			
1-32	A32	SOURCE USER IDENTIFIER	
33-64	A32	DESTINATION USER IDENTIFIER	
PREFACE DATA (PART 4):			
1-4	I4	CATEGORY CODE FOR DATA COMMUNICATION SESSION	
5-8	I4	CATEGORY CODE FOR INITIAL DISENGAGEMENT ATTEMPT IN SESSION	
9-12	I4	POINTER TO ORIGINATING USER	
13-16	I4	YEAR	REFERENCE TIME (DATE AT ORIGINATING USER SITE)
17-20	I4	MONTH	
21-24	I4	DAY	
25-28	I4	HOURS	REFERENCE TIME (LOCAL TIME-OF-DAY AT ORIGINATING USER SITE)
29-32	I4	MINUTES	
33-40	F8.0	SECONDS	
INITIAL STATE RECORD:			
1-4	I4	INITIAL COMMUNICATION STATE CODE FOR SOURCE USER	
5-8	I4	INITIAL COMMUNICATION STATE CODE FOR SOURCE HALF-SYSTEM	
EVENT RECORD:			
1-16	D16.0	EVENT TIME (SECONDS AFTER REFERENCE TIME)	
17-20	I4	COMMUNICATION STATE CODE FOR SOURCE USER	
21-24	I4	COMMUNICATION STATE CODE FOR SOURCE HALF-SYSTEM	
25-28	I4	REMOTE INTERFACE EFFECT CODE	
END-OF-HISTORY RECORD:			
1-16	D16.0	A NEGATIVE NUMBER	
17-20	I4	ZERO	
21-24	I4	ZERO	
25-28	I4	ZERO	

a. Source Overhead Information File

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS	
PREFACE DATA (PART 1):			
1-32	A32	FILE DESCRIPTOR	
PREFACE DATA (PART 2):			
1-64	A64	BATCH IDENTIFIER	
PREFACE DATA (PART 3):			
1-32	A32	SOURCE USER IDENTIFIER	
33-64	A32	DESTINATION USER IDENTIFIER	
PREFACE DATA (PART 4):			
1-4	I4	CATEGORY CODE FOR DATA COMMUNICATION SESSION	
5-8	I4	CATEGORY CODE FOR INITIAL DISENGAGEMENT ATTEMPT IN SESSION	
9-12	I4	POINTER TO ORIGINATING USER	
13-16	I4	YEAR	REFERENCE TIME (DATE AT ORIGINATING USER SITE)
17-20	I4	MONTH	
21-24	I4	DAY	
25-28	I4	HOURS	REFERENCE TIME (LOCAL TIME-OF-DAY AT ORIGINATING USER SITE)
29-32	I4	MINUTES	
33-40	F8.0	SECONDS	
INITIAL STATE RECORD:			
1-4	I4	INITIAL COMMUNICATION STATE CODE FOR DESTINATION HALF-SYSTEM	
5-8	I4	INITIAL COMMUNICATION STATE CODE FOR DESTINATION USER	
EVENT RECORD:			
1-16	D16.0	EVENT TIME (SECONDS AFTER REFERENCE TIME)	
17-20	I4	REMOTE INTERFACE EFFECT CODE	
21-24	I4	COMMUNICATION STATE CODE FOR DESTINATION HALF-SYSTEM	
25-28	I4	COMMUNICATION STATE CODE FOR DESTINATION USER	
END-OF-HISTORY RECORD:			
1-16	D16.0	A NEGATIVE NUMBER	
17-20	I4	ZERO	
21-24	I4	ZERO	
25-28	I4	ZERO	

b. Destination Overhead Information File

Figure 7. Record formats in overhead information files.

```
SOURCE OVERHEAD INFORMATION
NTIA - PDN test from Washington, DC                                    0915
NTIA - Term2 (NBS-Gaithersburg) NTIA - Host1 (Boulder)
   2   2   1  83  12  12  00  00   0.000
   1   1
000058259.389D+0   2   3   0
000058276.174D+0   3   2   0
000058276.307D+0   2   3   0
000058280.930D+0   3   2   0
000058281.063D+0   2   3   0
000058284.322D+0   3   2   0
000058284.455D+0   2   3   0
000058302.248D+0   3   2   0
000058302.381D+0   2   2   1
000058307.321D+0   3   2   0
000058307.603D+0   2   3   1
000058311.447D+0   3   2   0
000058312.327D+0   4   4   1
000058317.354D+0   5   4   0
000058317.487D+0   4   5   0
000058319.706D+0   5   4   0
000058319.839D+0   4   5   0
000058325.775D+0   1   1   0
000058381.373D+0   2   3   0
            ...
            ...
            ...
000060052.954D+0   1   1   0
000060108.522D+0   2   3   0
000060125.375D+0   3   2   0
000060125.508D+0   2   3   0
000060178.372D+0   2   4   0
000060178.372D+0   5   4   0
000060178.372D+0   1   1   0
000060239.535D+0   2   3   0
            ...
            ...
            ...
000060671.027D+0   1   1   0
      -1.000D+0   0   0   0
```

a. Source Overhead Information File

```
DESTINATION OVERHEAD INFORMATION
NTIA - PDN test from Washington, DC                                    0915
NTIA - Term2 (NBS-Gaithersburg) NTIA - Host1 (Boulder)
   2   2   1  83  12  12  00  00  00.000
   1   1
000058304.252D+0   0   2   3
000058305.974D+0   1   3   2
000058305.984D+0   0   2   3
000058306.117D+0   0   2   2
000058313.808D+0   0   2   3
000058313.941D+0   0   3   2
000058315.704D+0   0   4   5
000058316.032D+0   1   1   1
000058426.809D+0   0   2   3
            ...
            ...
            ...
000060043.127D+0   1   1   1
000060280.081D+0   0   2   3
            ...
            ...
            ...
000060290.222D+0   1   1   1
000060529.820D+0   0   2   3
000060530.502D+0   1   3   2
000060530.512D+0   0   2   3
000060530.645D+0   0   2   2
000060537.469D+0   0   2   3
000060537.602D+0   0   3   2
000060539.692D+0   0   4   5
000060540.172D+0   1   1   1
000060651.129D+0   0   2   3
000060651.836D+0   1   3   2
000060651.846D+0   0   2   3
000060651.979D+0   0   2   2
000060659.319D+0   0   2   3
000060659.452D+0   0   3   2
000060660.673D+0   0   4   5
000060661.156D+0   1   1   1
      -1.000D+0   0   0   0
```

b. Destination Overhead Information File

Figure 8. Examples of overhead information files.

Preface Data (Part 4). This record contains values for a set of session and batch descriptors. The category code for a data communication session is 1 if the sessions in a batch are connectionless and 2 if they are connection oriented. The category code for the initial disengagement attempt in a session is 1 if the attempt is independent and 2 if it is negotiated. The pointer to the originating user is 1 if the originating user is the source user and 4 if the originating user is the destination user. All event times in a performance data batch are expressed as seconds after a specified reference time. The reference time is given in this record as the local date and time-of-day at the originating user site. In the ITS implementation of ANS X3.141, the specified date is the date (at the originating user site) on which data extraction was initiated and the specified time-of-day is midnight (00:00:00.000) on that date. Batch and session descriptor values in the source overhead information file must be identical to those in the destination overhead information file.

Initial State Record. In each overhead information file, this record contains communication state codes for entities at the local user-system interface prior to the first event recorded in the file. Note that the user code precedes the half-system code in the source file and follows the half-system code in the destination file.

Event Record. Each performance-significant event observed at a monitored source or destination interface is represented by an event record in the respective overhead information file. An event record contains the event time (in seconds after the reference time described previously), the communication state codes for entities at the local interface subsequent to the event, and the remote interface effect code. In the source file, the half-system communication state code follows the user communication state code and is followed by the remote interface effect code. In the destination file, these three codes are written in the reverse order. The event records in an overhead information file must be arranged in the order of increasing event time.

End-of-History Record. This record is the last in an overhead information file. It has the same format as an event record, but contains a negative event time and zero values for other data items. It normally follows the last event record in the file; in exceptional cases, when an overhead information file does not contain any event record, the end-of-history record follows the initial state record. The negative event time informs the event history consolidation routine that the file does not contain additional event records.

3.2.3 User Information Files

The source user information file (SUI) and the destination user information file (DUI) in a performance data batch contain records of user information blocks transferred across the local user-system interface during the measurement period. Data recorded for each block include the relevant user information transfer event times and an ASCII-character representation of the binary content of the transferred (transmitted or received) user information.

The ASCII-character representation of the user information in a block is obtained as follows. The binary representation of the user information is divided into a sequence of 15-bit strings as shown in Figure 9. The last string in the block is completed, if necessary, with binary zero fill. Each string is regarded as the binary representation of a decimal integer, where the bit of lowest index is the most significant bit. The user information in a block is thus mapped into a sequence of decimal integers in the range 0-32,767. The digits for each integer are stored in a user information file as an ASCII-character string right-justified in a 5-character field with ASCII-zero fill on the left. The data in each field are read and converted by FORTRAN reduction routines using an I5 edit descriptor. After conversion, the low order (least significant) 15 bits of the corresponding storage location in memory are a replica of the original user information bit string. The length of the user information bit strings employed in the ASCII-character representation scheme (i) allows the reduction programs to run on computers that allocate 16-bit locations for storing integers and (ii) avoids storing user information in the sign bit of such locations.

The sequence of records in a user information file is shown in Figure 10. Formats and contents of individual records are summarized in Figure 11; additional details are provided in the paragraphs that follow. Examples of user information files are illustrated in Figure 12. User information files must conform to the specifications presented in this section. The source user information file is referenced in every reduction run, but the destination user information file is referenced only if user information transfer performance assessment is enabled.

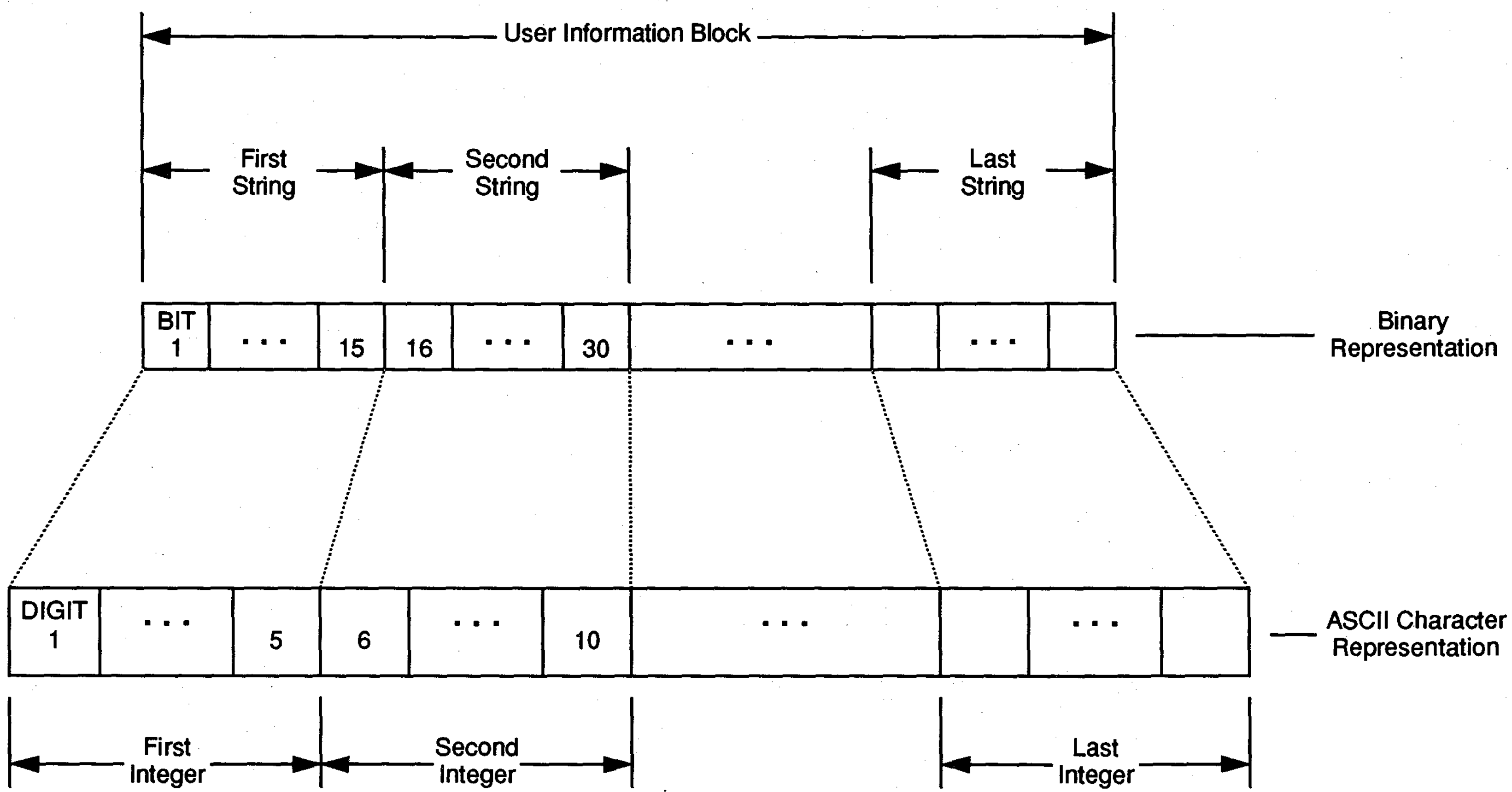


Figure 9. ASCII-character representation of user information.

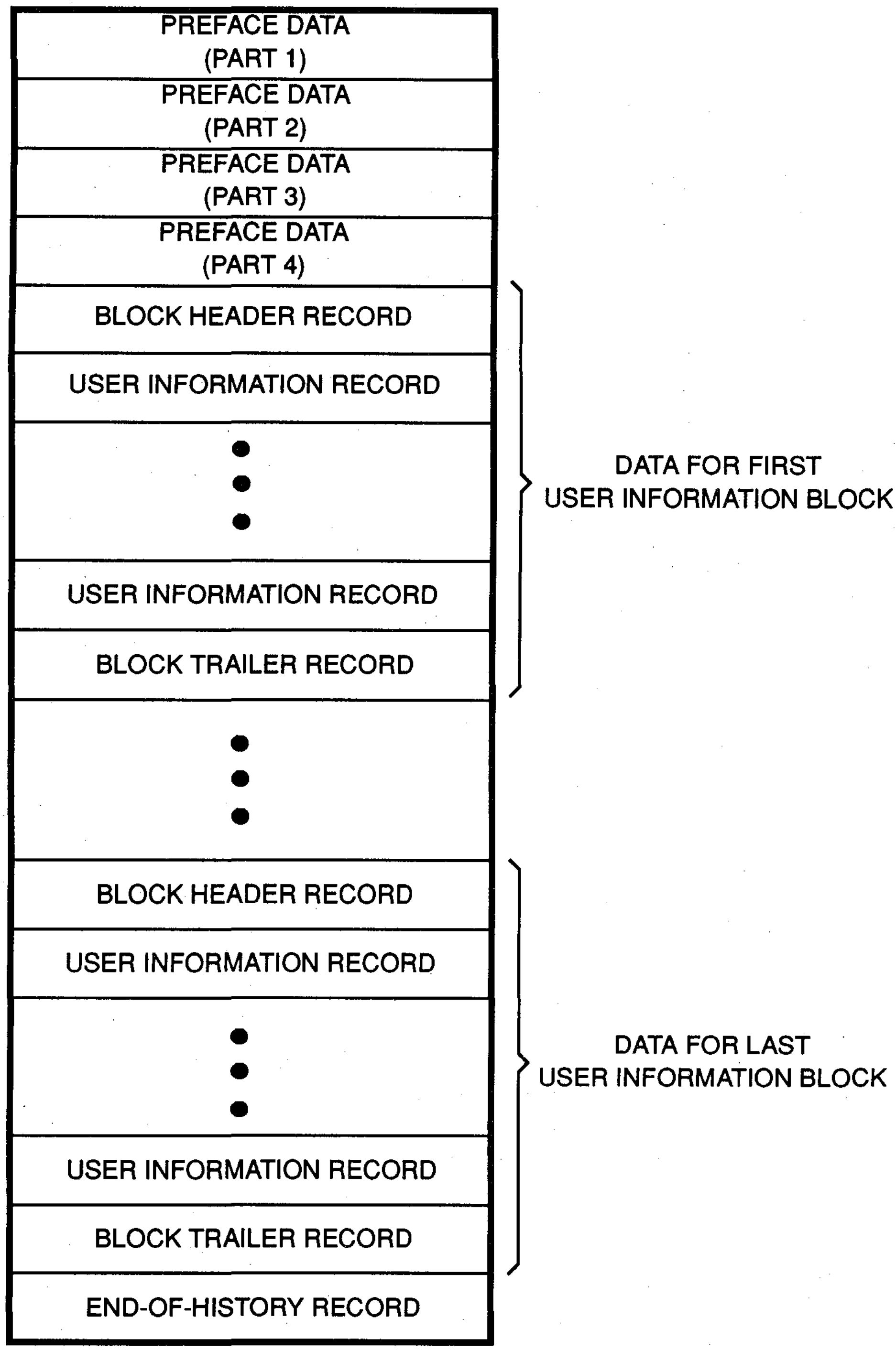


Figure 10. Record sequence in user information files.

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS	
PREFACE DATA (PART 1):			
1-32	A32	FILE DESCRIPTOR	
PREFACE DATA (PART 2):			
1-64	A64	BATCH IDENTIFIER	
PREFACE DATA (PART 3):			
1-32	A32	SOURCE USER IDENTIFIER	
33-64	A32	DESTINATION USER IDENTIFIER	
PREFACE DATA (PART 4):			
1-4	I4	YEAR	REFERENCE TIME (DATE AT ORIGINATING USER SITE)
5-8	I4	MONTH	
9-12	I4	DAY	
13-16	I4	HOURS	REFERENCE TIME (LOCAL TIME-OF-DAY AT ORIGINATING USER SITE)
17-20	I4	MINUTES	
21-28	F8.0	SECONDS	
BLOCK HEADER/TRAILER RECORD:			
1-8	F8.0	BLOCK INDEX	
9-16	F8.0	INITIAL BIT INDEX	
17-24	F8.0	BLOCK SIZE (BITS)	
25-40	D16.0	EVENT TIME FOR START OF BLOCK INPUT (SECONDS AFTER REFERENCE TIME)	
41-56	D16.0	EVENT TIME FOR START OF BLOCK TRANSFER (SECONDS AFTER REFERENCE TIME)	
USER INFORMATION RECORD:			
1-5	I5	USER INFORMATION FIELD	
6-10	I5	USER INFORMATION FIELD	
		• • •	
76-80	I5	USER INFORMATION FIELD	
END-OF-HISTORY RECORD:			
1-8	F8.0	ZERO OR A NEGATIVE NUMBER	
9-16	F8.0	ZERO	
17-24	F8.0	ZERO	
25-40	D16.0	ZERO	
41-56	D16.0	ZERO	

a. Source User Information File

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS	
PREFACE DATA (PART 1):			
1-32	A32	FILE DESCRIPTOR	
PREFACE DATA (PART 2):			
1-64	A64	BATCH IDENTIFIER	
PREFACE DATA (PART 3):			
1-32	A32	SOURCE USER IDENTIFIER	
33-64	A32	DESTINATION USER IDENTIFIER	
PREFACE DATA (PART 4):			
1-4	I4	YEAR	REFERENCE TIME (DATE AT ORIGINATING USER SITE)
5-8	I4	MONTH	
9-12	I4	DAY	
13-16	I4	HOURS	REFERENCE TIME (LOCAL TIME-OF-DAY AT ORIGINATING USER SITE)
17-20	I4	MINUTES	
21-28	F8.0	SECONDS	
BLOCK HEADER/TRAILER RECORD:			
1-8	F8.0	BLOCK INDEX	
9-16	F8.0	INITIAL BIT INDEX	
17-24	F8.0	BLOCK SIZE (BITS)	
25-40	D16.0	EVENT TIME FOR END OF BLOCK TRANSFER (SECONDS AFTER REFERENCE TIME)	
USER INFORMATION RECORD:			
1-5	I5	USER INFORMATION FIELD	
5-10	I5	USER INFORMATION FIELD	
		• • •	
76-80	I5	USER INFORMATION FIELD	
END-OF-HISTORY RECORD:			
1-8	F8.0	ZERO OR A NEGATIVE NUMBER	
9-16	F8.0	ZERO	
17-24	F8.0	ZERO	
25-40	D16.0	ZERO	

b. Destination User Information File

Figure 11. Record formats in user information files.

```
SOURCE USER INFORMATION
NTIA-ITS (Boulder)                                        1424
NTIA - terminal                  NTIA - host
  87   9  22  00  00  00.000
     1.       1.    1024.    36694.769D+0    36694.769D+0
06168044281866209587089052083700720167050886804433118870184706698187130321012359
09112075012634622262110571671308346123601118419476117270078908899186322781818742
13979034700320805525050101044919184279531347704366263762237221009280452827629026
09121239562717501270272820224907270152120978720632108310923819017237530270227473
15392231332711830579000000000000000000000000000000000000000000000000000000000000
     1.       1.    1024.    36694.769D+0    36694.769D+0
     2.    1025.    1024.    36694.897D+0    36694.897D+0
08747057882785422230250271885704206203420669805144279491311002634198890944021082
                                    ...
                                    ...
                                    ...
10297077050161301061020480000000000000000000000000000000000000000000000000000000
     9.    8193.    1024.    36695.845D+0    36695.845D+0
    10.    9217.    1024.    36695.978D+0    36695.978D+0
14008054651095717654171612604902184274760682006994105421314025274116842836821067
11049212750209209380067871352529396259411105723310284881410025530023172622014187
10680047652670005540274110238828820131141105303602186982168719153281290584019532
08866237621851025907091381255312496200650875605202270780504513122147572119027451
10781045582739821315000000000000000000000000000000000000000000000000000000000000
    10.    9217.    1024.    36695.978D+0    36695.978D+0
    11.   10241.    1024.    36696.112D+0    36696.112D+0
14762066102823729782150101172924716133831399222163021540080323458033412632225933
                                    ...
                                    ...
                                    ...
11315219170283026470225280000000000000000000000000000000000000000000000000000000
    18.   17409.    1024.    36697.821D+0    36697.821D+0
    19.   18433.    1024.    36697.965D+0    36697.965D+0
06324235700167621795127612587719624139041028921714273410582900898023850787813906
06587048771169205494232272191319114297681436203482284550603512898157932062425171
09756239562814201843024911967729892206021525603546263121414907034189130888225144
06827071291171913859110661063309972305121464020944268932587704547054451712427952
14524233250276100870266240000000000000000000000000000000000000000000000000000000
    19.   18433.    1024.    36697.965D+0    36697.965D+0
    20.   19457.    1024.    36698.097D+0    36698.097D+0
13480067382727225699170901072528902144191298423948115940975120954117372878419318
                                    ...
                                    ...
                                    ...
09783215980209305524122880000000000000000000000000000000000000000000000000000000
    79.   79873.    1024.    36712.406D+0    36712.406D+0
    80.   80897.    1024.    36712.538D+0    36712.538D+0
14519232511179001588168251886105810269880757921388283583037321169167970124617463
09010217782830301365169390970519086252090924220940198850573506546127252423428013
09882201721069514038044981784901252271870952222677198222160310706229442940629299
10043230011092026389193061580909396195101400421528108891763502691145490165014451
10668237091095401955081920000000000000000000000000000000000000000000000000000000
    80.   80897.    1024.    36712.538D+0    36712.538D+0
    -1.      0.       0.        0.000D+0        0.000D+0
```

a. Source User Information File

```
DESTINATION USER INFORMATION
NTIA-ITS (Boulder)                                        1424
NTIA - terminal                  NTIA - host
  87   9  22  00  00  00.000
     1.       1.    1024.    36695.554D+0
06168044281866209587089052083700720167050886804433118870184706698187130321012359
09112075012634622262110571671308346123601118419476117270078908899186322781818742
13979034700320805525050101044919184279531347704366263762237221009280452827629026
09121239562717501270272820224907270152120978720632108310923819017237530270227473
15392231332711830579000000000000000000000000000000000000000000000000000000000000
     1.       1.    1024.    36695.554D+0
     2.    1025.    1024.    36695.734D+0
08747057882785422230250271885704206203420669805144279491311002634198890944021082
                                    ...
                                    ...
                                    ...
10297077050161301061020480000000000000000000000000000000000000000000000000000000
     9.    8193.    1024.    36697.778D+0
    10.    9217.    1024.    36697.962D+0
14008054651095717654171630243718568274760682006994105421314025274116842836821067
11049212750209209380067871352529396259411105723310284881410025530023172622014187
10680047652670005540274110238828820131141105303602186982168719153281290584019532
08866237621851025907091381255312496200650875605202270780504513122147572119027451
10781045582739821315000000000000000000000000000000000000000000000000000000000000
    10.    9217.    1024.    36697.962D+0
    11.   10241.    1024.    36698.123D+0
14762066102823729782150101172924716133831399222163021540080323458033412632225933
                                    ...
                                    ...
                                    ...
11315219170283026470225280000000000000000000000000000000000000000000000000000000
    18.   17409.    1024.    36699.379D+0
    19.   18433.    1024.    36699.587D+0
06324235721811913204110582069708352254270964306747025740106108827032890931830284
06838226132801030550130982196904320133661375620062099291765425410085850988014710
10679235802615030515153150236929934123431372304317026692999117034259922877822127
12983038001809229862047632710024804249030861207060108230162102827188682480231048
06706233250276100870266240000000000000000000000000000000000000000000000000000000
    19.   18433.    1024.    36699.587D+0
    20.   19457.    1024.    36700.482D+0
13480067382727225699170901072528902144191298423948115940975120954117372878419318
                                    ...
                                    ...
                                    ...
09783215980209305524122880000000000000000000000000000000000000000000000000000000
    79.   79873.    1024.    36714.662D+0
    80.   80897.    1024.    36714.851D+0
14519232511179001588168251886105810269880757921388283583037321169167970124617463
09010217782830301365169390970519086252090924220940198850573506546127252423428013
09882201721069514038044981784901252271870952222677198222160310706229442940629299
10043230011092026389193061580909396195101400421528108891763502691145490165014451
10668237091095401955081920000000000000000000000000000000000000000000000000000000
    80.   80897.    1024.    36714.851D+0
    -1.      0.       0.        0.000D+0
```

b. Destination User Information File

Figure 12. Examples of user information files.

Preface Data (Part 1). This record contains a file descriptor consisting of the character string SOURCE USER INFORMATION in the source user information file and the string DESTINATION USER INFORMATION in the destination user information file. Each descriptor is left-justified in a 32-character field with blank fill on the right. During the preliminary examination of input data in PROLOG, the descriptor field read from the SUI or DUI file is compared with the prescribed (expected) descriptor field. Any difference between the two fields is regarded as a fatal input data error and further processing in the reduction run is suppressed.

Preface Data (Part 2). This record contains the batch identifier described in the preceding section and is identical to its counterpart in the corresponding overhead information file.

Preface Data (Part 3). This record contains the source and destination user identifiers described in the preceding section and is identical to its counterpart in the corresponding overhead information file.

Preface Data (Part 4). This record contains the reference time described in the preceding section. Each date and time-of-day data item is identical to its counterpart in the overhead information files.

Block Header Record. For each block in a user information file, this record contains values for a set of block descriptors and values of relevant user information transfer event times. Evaluated block descriptors are

- Block Index - the ordinal number associated with the block when all user information blocks recorded in the file are arranged in the order of their transfer across the local user-system interface,
- Initial Bit Index - the ordinal number associated with the initial bit in the block when all user information bits in the file are arranged according to the order induced by combining (i) the chronological order of all blocks in the file and (ii) the bit order within each block (the order used to divide the block into 15-bit strings as described earlier), and
- Block Size - the number of user information bits in the block.

The header record for a block in the source user information file contains event times (in seconds after the reference time) for the start of input to the system and the start of transfer. The header record for a block in the destination user information file contains the event time for the end of transfer.

User Information Record. One or more of these records follow the header record for a user information block and contain the ASCII-character representation of the block described earlier in this section. Each user information field contains the ASCII digits for the decimal integer equivalent of 15 successive user information bits. The decimal digits are right-justified in the field with ASCII zero fill on the left. A user information record contains 16 such fields and represents a maximum of 240 user information bits. If necessary, unused fields in the last record for a block are filled with ASCII zeros. When analyzing data in a user information record, reduction programs ignore any fill that follows the last user information bit in the block.

Block Trailer Record. For each block in a user information file, this record follows the associated user information records and is identical to the block header record. It provides data that enable reduction routines to backspace in a user information file.

End-of-History Record. This record is the last in a user information file. It has the same format as a block header (or trailer) record, but contains a negative or zero block index and zero values for other data items. It normally follows the trailer record for the last block in the file; in exceptional cases, when the file does not contain any user information, the end-of-history record follows the preface records. A zero or negative block index informs reduction routines that the file does not contain data for additional blocks.

3.3 Examination of Input Data

Subroutine **DATXAM** implements the examination of input data files for compliance with certain format and content requirements presented in Sections 3.1 and 3.2. Subroutine **CKSPEC** checks the specifications input file and subroutine **CKINFO** checks the overhead and user information files. Error conditions that may be observed by these routines are listed in this section. Other error conditions, in which the format of an input data record differs from that required by the READ and FORMAT statements used by the examination routine, may result in a system I/O error.

If **CKSPEC** or **CKINFO** observes an error, the routine executes a procedure that equates a processing status code to 1, writes an appropriate diagnostic to the assessment summary file (**SUM**), closes all files opened by the routine, and returns control to the calling program. The reduction run is then called a suspended run and further data processing in the run is suppressed. Otherwise, the run is called a normal reduction run and the value of the processing status code is 0.

3.3.1 Examination of Specifications Input File

Subroutine **CKSPEC** examines the **SPI** file for the following error conditions:

- the file descriptor read from the **SPI** file does not match the prescribed descriptor for the specifications input file,
- an assessment option read from the **SPI** file is not 0 (suppress assessment) or 1 (enable assessment), and
- all assessment options read from the **SPI** file are 0.

If the assessment options are valid, **CKSPEC** reads reduction specifications records from the **SPI** file in accordance with the observed option values and the record formats defined in Section 3.1. No data in these records are examined by **CKSPEC**.

3.3.2 Examination of Performance Data Files

Subroutine **CKINFO** examines both preface and performance data in the overhead and user information files. The routine first examines the SOI file and checks preface data for the following error conditions:

- the file descriptor read from the **SOI** file does not match the prescribed descriptor for the source overhead information file,
- the batch identifier read from the **SOI** file does not match that read from the **SPI** file,
- the session category code read from the **SOI** file is not 1 (connectionless) or 2 (connection oriented),
- the disengagement category code read from the **SOI** file is not 1 (independent) or 2 (negotiated),
- the session category code read from the **SOI** file is 1 (connectionless) and the disengagement category code read from the file is 2 (negotiated),
- the originating user pointer read from the **SOI** file is not 1 (source) or 4 (destination), and

- the session category code read from the **SOI** file is 1 (connectionless) and the originating user pointer read from the file is 4 (destination).

CKINFO examines the initial state record in the **SOI** file for two error conditions:

- the initial communication state code for the user or the adjacent half-system is not in the range 0-5 and
- the initial communication state code for the user is 3 (committed-active) or 5 (closing-active) and the code for the adjacent half-system is also 3 or 5. (At an interface, no more than one non-idle entity may be in the active ancillary state at any given time.)

CKINFO checks event records in the **SOI** file for the following error conditions:

- the event time in the current record is earlier than the event time in the preceding record,
- the communication state code for the user or the adjacent half-system is not in the range 0-5,
- the communication state code for the user is 3 or 5 and the code for the adjacent half-system is also 3 or 5, and
- the remote interface effect code is not 0 (effect absent) or 1 (effect present).

If no error is observed in the **SOI** file, **CKINFO** next examines the **DOI** file. The routine compares the file descriptor and the batch identifier read from the **DOI** file with, respectively, the prescribed descriptor for the destination overhead information file and the batch identifier read from the **SPI** file. Subsequent preface data are checked by comparing each data item read from the **DOI** file with the corresponding item read from the **SOI** file. The initial state and event records in the **DOI** file are checked for the same error conditions as the corresponding records in the **SOI** file.

While examining the overhead information files, **CKINFO** identifies start and end times for the measurement period associated with the performance data batch. The <u>measurement period start time</u> is the earliest of the event times recorded in the source and destination overhead information files; <u>the measurement period end time</u> is the latest of these event times.

If no error is observed in the overhead information files, **CKINFO** continues by checking the **SUI** file. Preface data are examined for the following error conditions:

- the file descriptor read from the **SUI** file does not match the prescribed descriptor for the source user information file,
- the batch identifier read from the **SUI** file does not match that read from the **SPI** file,
- the source user identifier read from the **SUI** file does not match that read from the **SOI** file,
- the destination user identifier read from the **SUI** file does not match that read from the **SOI** file,
- the date component of reference time read from the **SUI** file does not match that read from the **SOI** file, and
- the time-of-day component of reference time read from the **SUI** file does not match that read from the **SOI** file.

CKINFO checks block header records in the **SUI** file for the following error conditions:

- the input start time for the current block is later than the transfer start time,
- the transfer start time for the current block is not later than the transfer start time for the preceding block,
- the input start time for the first block in the file is earlier than the start of the source overhead event history, and
- the transfer start time for the current block is later than the end of the source overhead event history.

If no error is observed in a block header record, **CKINFO** reads the associated user information and block trailer records, but does not check any of their data.

CKINFO examines the **DUI** file only if user information transfer performance assessment is enabled in the reduction run. (Reduction programs do not require data from the **DUI** file when user information transfer performance assessment is suppressed.) The routine compares the file descriptor and the batch identifier read from the **DUI** file with, respectively, the prescribed descriptor for the destination user information file and the batch identifier read from the **SPI**

file. Subsequent preface data are checked by comparing the data items read from the DUI file with the corresponding items read from the SOI file. CKINFO checks block header records in the DUI file for the following error conditions:

- the transfer end time for the current block is earlier than the transfer end time for the preceding block,
- the transfer end time for the first block in the file is earlier than the start of the destination overhead event history, and
- the transfer end time for the current block is later than the end of the destination overhead event history.

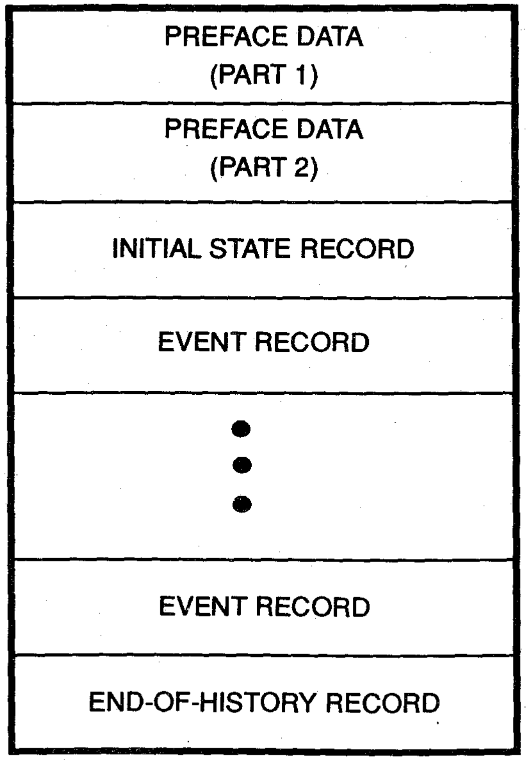
If no error is observed in a block header record, CKINFO reads the associated user information and block trailer records, but does not check any of their data.

Note that CKINFO requires transfer start times for successive blocks in the SUI file to be strictly increasing. This requirement ensures that defined input/output times for selected availability or throughput transfer samples exceed zero and enables the user information bit transfer rates for such samples to be evaluated. Note also that CKINFO requires all event times recorded in a user information file to be in the period spanned by the corresponding overhead event history.

3.4 Consolidation of Reference Event Data

In a normal reduction run, subroutine CONREV combines reference event data in the source and destination overhead information files to produce a unified history of primary access and disengagement reference events and all ancillary reference events observed at the monitored user-system interfaces during the measurement period. This event history is written to the consolidated overhead information file (COI).

The sequence of records in the consolidated overhead information file is shown in Figure 13a; formats and contents of individual records are summarized in Figure 13b. Figure 14 shows an edited portion of the consolidated overhead information file obtained by combining reference event data in the source and destination overhead information files in Figure 8. Each event recorded in the source or destination overhead information file is represented by an event record in the COI file. Event records in the COI file are arranged in chronological order. (If an event in the source overhead information file occurs at the same



a. Record Sequence

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
PREFACE DATA (PART 1):		
1-32	A32	FILE DESCRIPTOR
PREFACE DATA (PART 2):		
1-64	A64	BATCH IDENTIFIER
INITIAL STATE RECORD:		
1-4	I4	INITIAL COMMUNICATION STATE CODE FOR SOURCE USER
5-8	I4	INITIAL COMMUNICATION STATE CODE FOR SOURCE HALF-SYSTEM
9-12	I4	INITIAL COMMUNICATION STATE CODE FOR DESTINATION HALF-SYSTEM
13-16	I4	INITIAL COMMUNICATION STATE CODE FOR DESTINATION USER
EVENT RECORD:		
1-8	F8.0	EVENT INDEX
9-24	D16.9	EVENT TIME (SECONDS AFTER REFERENCE TIME)
25-28	I4	COMMUNICATION STATE CODE FOR SOURCE USER
29-32	I4	COMMUNICATION STATE CODE FOR SOURCE HALF-SYSTEM
33-36	I4	COMMUNICATION STATE CODE FOR DESTINATION HALF-SYSTEM
37-40	I4	COMMUNICATION STATE CODE FOR DESTINATION USER
END-OF-HISTORY RECORD:		
1-8	F8.0	A NEGATIVE NUMBER
9-24	D16.9	ZERO
25-28	I4	ZERO
29-32	I4	ZERO
33-36	I4	ZERO
37-40	I4	ZERO

b. Record Formats

Figure 13. Record sequence and record formats in consolidated overhead information file.

```
CONSOLIDATED OVERHEADINFORMATION
NTIA - PDN test from Washington, DC                                 0915
   1    1    1    1
       1.   .582593890E+05    2    3    1    1 ... ACCESS REQUEST
       2.   .582761740E+05    3    2    1    1       (Start of Session)
       3.   .582763070E+05    2    3    1    1
       4.   .582809300E+05    3    2    1    1
       5.   .582810630E+05    2    3    1    1
       6.   .582843220E+05    3    2    1    1
       7.   .582844550E+05    2    3    1    1
       8.   .583022480E+05    3    2    1    1
       9.   .583023810E+05    2    2    3    1
      10.   .583042520E+05    2    2    2    3 ... NONORIGINATING USER COMMITMENT
      11.   .583059740E+05    2    3    3    2
      12.   .583059840E+05    2    3    2    3
      13.   .583061170E+05    2    3    2    2
      14.   .583073210E+05    3    2    2    2
      15.   .583076030E+05    2    3    3    2
      16.   .583114470E+05    3    2    3    2
      17.   .583123270E+05    4    4    3    2 ... SOURCE DISENGAGEMENT REQUEST
      18.   .583138080E+05    4    4    2    3
      19.   .583139410E+05    4    4    3    2
      20.   .583157040E+05    4    4    4    5
      21.   .583160320E+05    4    5    1    1 ... DESTINATION DISENGAGEMENT
      22.   .583173540E+05    5    4    1    1       CONFIRMATION
      23.   .583174870E+05    4    5    1    1
      24.   .583197060E+05    5    4    1    1
      25.   .583198390E+05    4    5    1    1
      26.   .583257750E+05    1    1    1    1 ... SOURCE DISENGAGEMENT CONFIRMATION
                                                    (End of Session)
      27.   .583813730E+05    2    3    1    1 ... ACCESS REQUEST
                       ...                          (Start of Session)
                       ...
                       ...
     370.   .600529540E+05    1    1    1    1 ... SOURCE DISENGAGEMENT CONFIRMATION
                                                    (End of Session)
     371.   .601085220E+05    2    3    1    1 ... ACCESS REQUEST
     372.   .601253750E+05    3    2    1    1       (Start of Session)
     373.   .601255080E+05    2    3    1    1
     374.   .601783720E+05    2    4    1    1 ...       SYSTEM
     375.   .601783720E+05    5    4    1    1 ... BLOCKING  SIGNAL
     376.   .601783720E+05    1    1    1    1 ... (End of Session)

     377.   .602395350E+05    2    3    1    1 ... ACCESS REQUEST
                       ...                          (Start of Session)
                       ...
                       ...
     460.   .606710270E+05    1    1    1    1 ... SOURCE DISENGAGEMENT CONFIRMATION
                                                    (End of Session)
      -1.   .000000000E+00    0    0    0    0 ... (End-of-History Record)
```

Figure 14. Event records in a typical consolidated event history.

time as an event in the destination overhead information file, the record of the source event precedes that of the destination event in the consolidated event history.)

The scheme employed by **CONREV** to determine communication states subsequent to an event incorporates the ancillary event history consolidation procedure specified in ANS X3.141. This scheme is outlined in Figure 15. States of entities at the local interface (the interface where the event occurs) are those given in the corresponding event record in the source or destination overhead information file. States of entities at the remote interface subsequent to an event are jointly determined by states of the remote entities prior to the event and by the remote interface effect code associated with the event. The communication state of the remote half-system is changed by an event if the associated remote interface effect code is 1 and (i) each remote entity is in an idle (0 or 1) state prior to the event or (ii) each remote entity is in the committed-waiting (2) state or the closing-waiting (4) state prior to the event. The remote half-system undergoes a transition to the committed-active (3) state in case (i) and to the associated active (3 or 5) state in case (ii). Otherwise, the communication state of the remote half-system is unchanged by the event. The communication state of the remote user is unchanged in all cases.

Event 9 in Figure 14 illustrates case (i). This event corresponds to the ninth event in the source overhead information file (Figure 8a), for which the associated remote interface effect code is 1. Before the event, both remote (destination) entities are in the idle-active (1) state; after the event, the remote half-system is in the committed-active (3) state. Event 11 in Figure 14 illustrates case (ii). This event corresponds to the second event in the destination overhead information file (Figure 8b), for which the associated remote interface effect code is also 1. Before the event, both remote (source) entities are in the committed-waiting (2) state; after the event, the remote half-system is in the committed-active (3) state. Another example of case (ii) is provided by event 21 in Figure 14.

In Figure 14, comments on the right (not contained in the **COI** file) identify the performance significance of key primary state transitions. Note that the System Blocking Signal is represented by two event records (374 and 375) as described in Section 3.2.1. Blank lines have been editorially inserted in the event history to display session boundaries.

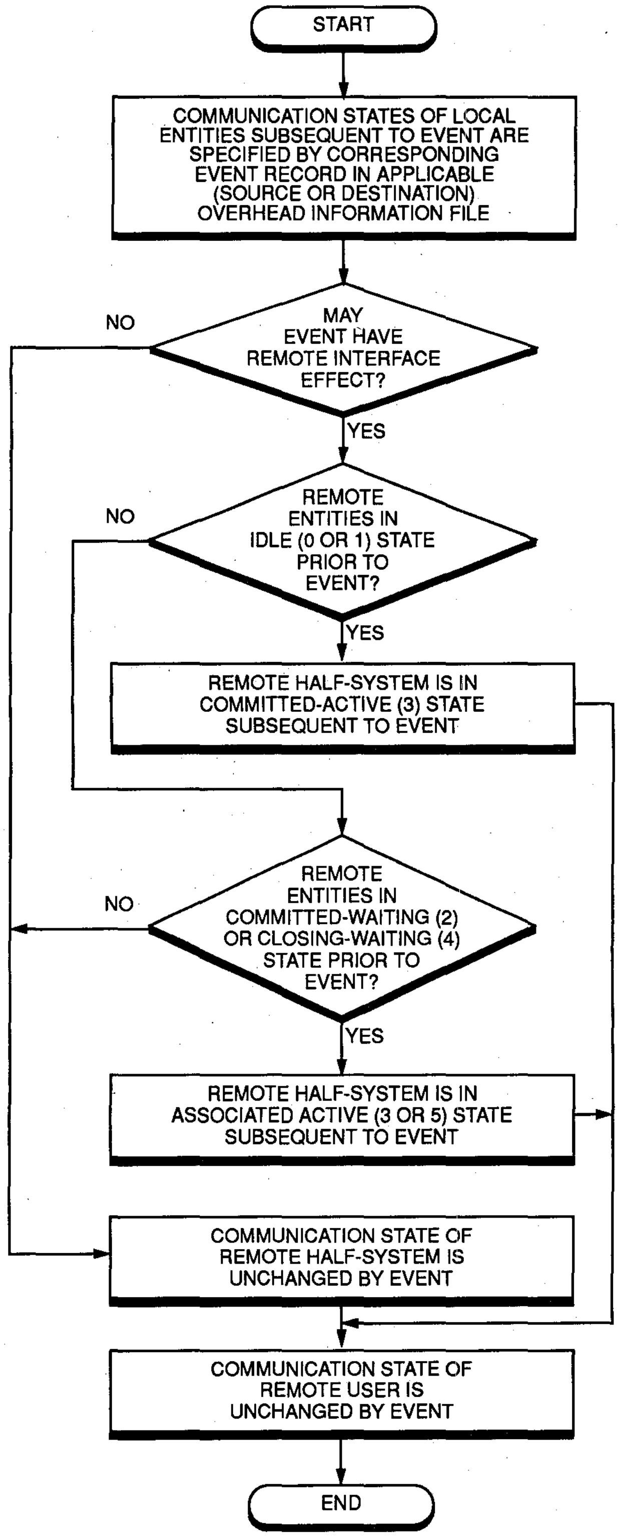


Figure 15. Determination of communication states in the consolidated event history.

4. PERFORMANCE ASSESSMENT

To assist in the interpretation of results produced by the NTIA implementation of ANS X3.141, this section discusses key concepts used by performance assessment procedures in program **ANALYZ** to identify individual performance trials and determine their outcomes. Concepts used in access, user information transfer, and disengagement performance assessment procedures are discussed in Sections 4.1, 4.2, and 4.3, respectively. Concepts used in performance time allocation procedures are discussed in Section 4.4.

4.1 Access Performance Assessment

In a normal reduction run for which access performance assessment is enabled, procedures in subroutine **ACCESS** identify access attempts recorded in a performance data batch and determine their outcomes.

Input to access performance assessment procedures consists of

- the event history in the consolidated overhead information file (**COI**),
- the event history in the source user information file (**SUI**), and
- specified values of Access Time and User Fraction of Access Time used to determine outcomes of access attempts.

The specified values indicated above are obtained from the consolidated specifications file (**CSP**). Outcomes of individual access attempts are recorded in the <u>access outcome file</u> (**ACO**).

Procedures for identifying access attempts are discussed in Section 4.1.1 and procedures for determining their outcomes are discussed in Section 4.1.2.

4.1.1 Identification of Access Attempts

Subroutine **ACCESS** identifies the start of an access attempt and the end of the associated performance period. The start of an access attempt always corresponds to an Access Request event. An Access Request is represented in the consolidated overhead information file by an event record in which the originating user and the adjacent half-system undergo transitions from the idle-active (1) state to a committed (2 or 3) state. Records of three Access Requests are illustrated in Figure 14.

The end of an access performance period corresponds to whichever occurs first: the end of the access attempt or the end of the maximum performance period associated with the attempt. In accordance with ANS X3.102, the length of the maximum performance period for an access attempt is three times the specified value of Access Time. Throughout this section, the end of the maximum performance period for an access, block transfer, or disengagement attempt is called the performance deadline for the attempt. Access timeout occurs if the end of an access attempt does not occur on or before the associated performance deadline.

The end of an access attempt normally corresponds to one of the following: the subsequent start of user information transfer, a System Blocking Signal, or a User Blocking Signal. The start of user information transfer in a session is represented in the source user information file by the earliest Start of Block Input event that is later than the Access Request for the session. A System Blocking Signal is represented in the consolidated overhead information file by two successive event records that contain the same event time. The first of these records corresponds to a transition from a committed (2 or 3) state to a closing (4 or 5) state by the half-system adjacent to the originating user and the second corresponds to the same primary state transition by the originating user. A System Blocking Signal is illustrated in Figure 14 by event records 374 and 375. A User Blocking Signal is similarly represented by two successive event records. In this case, the first of the two records corresponds to a transition from a committed state to a closing state by the issuing user and the second corresponds to the same primary state transition by the adjacent half-system.

Subsequent to an Access Request, the originating user and the adjacent half-system are in a committed state. Any event in which an entity at a relevant interface undergoes a transition from a committed state is regarded by **ACCESS** as the end of the access attempt if (i) the event is the earliest such transition that follows the Access Request and (ii) the transition is not preceded in the session by a Start of Block Input event (the start of user information transfer). The indicated transition is called a blocking event; a blocking event is normal if it is associated with a System Blocking Signal or with a User Blocking Signal issued at a relevant interface, and is anomalous otherwise. Only the originating user interface is relevant in a connectionless session, whereas both interfaces are relevant in a connection-oriented session.

In performance assessment procedures implemented by **ACCESS**, the end of an access performance period thus corresponds to whichever occurs first after the Access Request:

- Start of Block Input event,
- a blocking event, or
- the associated performance deadline.

4.1.2 Determination of Access Outcomes

Subroutine **ACCESS** determines outcomes of access attempts in accordance with the definitions given in ANS X3.102. These outcomes may be characterized as follows:

> Successful Access occurs in a connectionless session if user information transfer begins no later than the access performance deadline. Successful Access occurs in a connection-oriented session if user information transfer begins no later than the access performance deadline and the nonoriginating user is committed to the session prior to the start of transfer.
>
> Incorrect Access occurs in a connection-oriented session if user information transfer begins no later than the access performance deadline and the nonoriginating user is not committed to the session prior to the start of transfer. Incorrect Access does not occur in a connectionless session.
>
> Access Denial occurs if (i) a System Blocking Signal occurs no later than the access performance deadline or (ii) access timeout occurs, there is a system response on or before the access performance deadline, and the measured user fraction of performance time for the period does not exceed the specified value of User Fraction of Access Time.
>
> Access Outage occurs if there is no system response to the Access Request on or before the access performance deadline.
>
> User Blocking occurs if (i) a User Blocking Signal occurs no later than the access performance deadline or (ii) access timeout occurs, there is a system response on or before the access performance deadline, and the measured user fraction of performance time for the period exceeds the specified value of User Fraction of Access Time.

In accordance with ANS X3.102, an access attempt whose outcome is User Blocking is excluded from the set of trials used to estimate values of access performance parameters.

4.1.2.1 Outcome Determination Scheme

The scheme used in subroutine **ACCESS** to determine outcomes of access attempts is outlined by the flowchart in Figure 16. Procedures that identify the end of the performance period for an access attempt assign the attempt to one of the following categories: (i) access attempts in which the performance period is terminated by the start of user information transfer, (ii) access attempts in which the performance period is terminated by a blocking event, or (iii) access attempts in which the performance period is terminated by access timeout (the associated performance deadline). Each category corresponds to a column of decision symbols in Figure 16.

In a connectionless session (one whose category code is 1), the outcome of an access attempt in which the performance period is terminated by the start of user information transfer is Successful Access. In a connection-oriented session, the outcome of such an access attempt depends on the communication state of the nonoriginating user just prior to the start of transfer. If that user is in a committed (2 or 3) state, the outcome is Successful Access; otherwise, the outcome is Incorrect Access.

The outcome of an access attempt in which the performance period is terminated by a blocking event (a transition from the committed state by some entity at a relevant interface) depends on the nature of the blocking event. If the blocking event is associated with a System Blocking Signal, the outcome of the access attempt is Access Denial. If the blocking event is associated with a User Blocking signal issued at a relevant interface, the outcome is User Blocking. If the blocking event is anomalous, the outcome of the access attempt is not classified (the event history is anomalous or erroneous). An access attempt in which the performance period is terminated by an anomalous blocking event is excluded from the set of trials used to estimate values of access performance parameters.

A blocking event is associated with a System Blocking Signal if

- the event is a transition from a committed (2 or 3) state to a closing (4 or 5) state by the half-system adjacent to the originating user,

- the next event record corresponds to the same primary state transition by the originating user, and

- the two records contain the same event time.

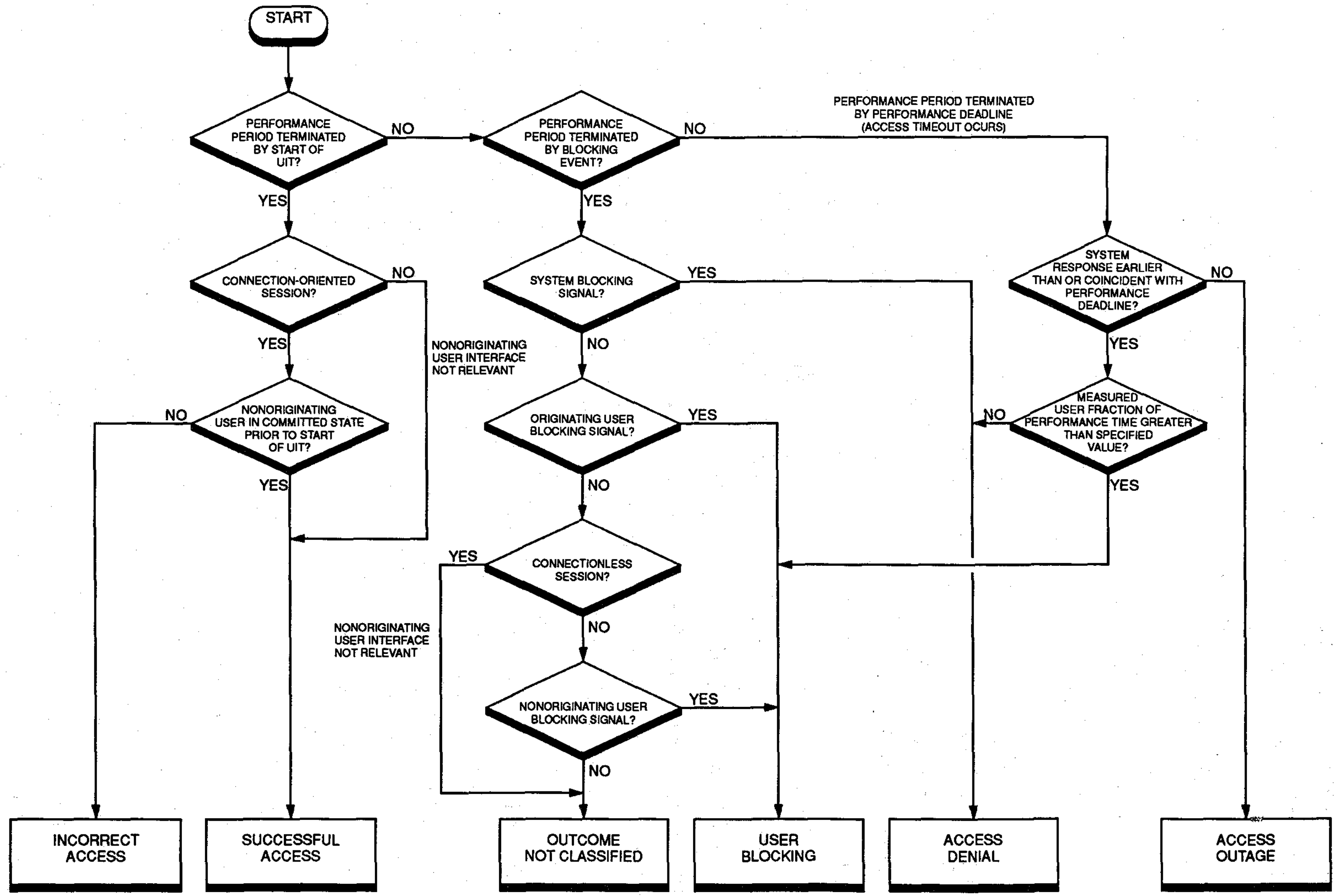


Figure 16. Scheme for determining access outcomes.

If the first of these criteria is satisfied and either of the last two is not, the blocking event is anomalous. A blocking event is associated with a User Blocking Signal issued by the originating user if

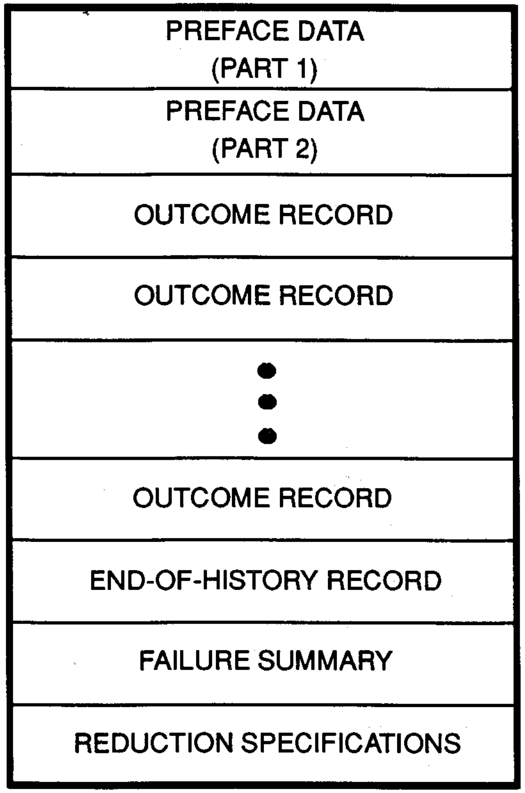
- the event is a transition from a committed (2 or 3) state to a closing (4 or 5) state by the originating user,
- the next event record corresponds to the same primary state transition by the adjacent half-system, and
- the two records contain the same event time.

As before, if the first of these criteria is satisfied and either of the last two is not, the blocking event is anomalous. In a connection-oriented session, analogous criteria are used to determine if a blocking event is associated with a User Blocking Signal issued by the nonoriginating user (the nonoriginating user cannot issue a Blocking Signal in a connectionless session).

The outcome of an access attempt in which the performance period is terminated by access timeout depends on the absence or presence of a system response to the Access Request during the performance period. (The first event that follows an Access Request in the consolidated event history is regarded by subroutine **ACCESS** as a system response to the request.) If there is no system response prior to or coincident with the performance deadline, the outcome of the access attempt is Access Outage. If a response occurs on or before the deadline, the routine evaluates the user fraction of performance time for the period. If the measured fraction exceeds the specified value for User Fraction of Access Time, responsibility for the excessive delay is attributed to user nonperformance and the outcome of the access attempt is User Blocking. Otherwise, responsibility for the delay is attributed to system nonperformance and the outcome is Access Denial.

4.1.2.2 Access Outcome File

Outcomes of individual access attempts are recorded in the access outcome file (**ACO**) as they are determined. The record sequence in this file is shown in Figure 17a, record formats are defined in Figure 17b, and an example of an access outcome file is shown in Figure 18. Each access attempt identified by subroutine **ACCESS** is represented by an <u>outcome record</u> in the access outcome file. The outcome record for an access attempt whose outcome is Successful Access contains



a. Record Sequence

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
PREFACE DATA (PART 1):		
1-32	A32	FILE DESCRIPTOR
PREFACE DATA (PART 2):		
1-64	A64	BATCH IDENTIFIER
OUTCOME RECORD (SUCCESSFUL PERFORMANCE):		
1-8	F8.3	OVERALL PERFORMANCE TIME FOR ACCESS ATTEMPT (SECONDS)
9-16	F8.3	USER PERFORMANCE TIME FOR ACCESS ATTEMPT (SECONDS)
OUTCOME RECORD (UNSUCCESSFUL PERFORMANCE):		
1-8	F8.0	OUTCOME CODE FOR UNSUCCESSFUL ACCESS ATTEMPT
9-16	F8.0	OUTCOME CODE FOR UNSUCCESSFUL ACCESS ATTEMPT
END-OF-HISTORY RECORD:		
1-8	F8.0	END-OF-HISTORY CODE (-30)
9-16	F8.0	END-OF-HISTORY CODE (-30)
FAILURE SUMMARY:		
1-8	F8.0	NUMBER OF ACCESS ATTEMPTS IN MEASUREMENT SAMPLE
9-16	F8.0	NUMBER OF 'INCORRECT ACCESS' OUTCOMES
17-24	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'INCORRECT ACCESS' OUTCOMES
25-32	F8.0	NUMBER OF 'ACCESS DENIAL' OUTCOMES
33-40	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'ACCESS DENIAL' OUTCOMES
41-48	F8.0	NUMBER OF 'ACCESS OUTAGE' OUTCOMES
49-56	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'ACCESS OUTAGE' OUTCOMES
REDUCTION SPECIFICATIONS:		
1-16	E16.8	SPECIFIED VALUE OF ACCESS TIME (SECONDS)
17-32	E16.8	SPECIFIED VALUE OF USER FRACTION OF ACCESS TIME

b. Record Formats

Figure 17. Record sequence and record formats in access outcome file.

```
ACCESS OUTCOME
NTIA - PDN test from Washington, DC                                   0915
  48.214   2.536
  47.348   1.686
  42.453   1.603
  47.594   1.606
  48.110   1.608
  43.132   1.604
  45.905   1.606
  44.096   1.606
  46.770   1.606
  42.995   1.606
  45.753   1.600
     -2.      -2.
  48.262   1.604
  43.788   1.601
  41.638   1.602
     -2.      -2.
  42.472   1.521
     -2.      -2.
  46.857   1.496
  47.718   1.521
    -30.     -30.
     20.       0.       0.       3.       0.       0.       0.
  0.45000000E+02  0.50000001E-01
```

Figure 18. Example of an access outcome file.

overall and user performance times, whereas the record for an unsuccessful access attempt contains a negative access outcome code. Incorrect Access, Access Denial, Access Outage, and User Blocking outcomes are indicated by -1, -2, -3, and -5, respectively. Access attempts whose outcomes are not classified are indicated by -9. The final outcome record in the file is followed by an end-of-history record. The latter contains an end-of-history code (-30) and has the same format as the outcome record for an unsuccessful access attempt. The subsequent failure summary record lists the number of trials in the measurement sample and lists, for each system-responsible failure category, the observed numbers of failures and pairs of consecutive failures. The final reduction specifications record in the access outcome file contains specified values of Access Time and User Fraction of Access Time used in outcome determination. Information in the access outcome file enables the statistical analysis program **STAR** (described in Volume 5 of this report) to calculate estimated values and their confidence limits for all access performance parameters defined in ANS X3.102.

4.2 User Information Transfer Performance Assessment

In a normal reduction run for which user information transfer performance assessment is enabled, procedures in subroutine **TRANSF**

- identify bit and block transfer attempts recorded in a performance data batch and determine their outcomes (misdelivery performance is not evaluated),
- select availability transfer samples for the measurement of Transfer Denial Probability and determine their outcomes, and
- select a throughput transfer sample for the measurement of long-term throughput parameters (User Information Bit Transfer Rate and User Fraction of Input/Output Time).

Input to user information transfer performance assessment procedures consists of

- the event history in the consolidated overhead information file (COI),
- event histories in the source user information file (SUI) and the destination user information file (DUI),

- specified values of Block Transfer Time and User Fraction of Block Transfer Time used to determine outcomes of block transfer attempts,
- values of transfer availability measurement specifications, and
- values of correlator specifications.

All specified values indicated above are obtained from the consolidated specifications file (**CSP**). Transfer availability measurement specifications consist of the minimum number of bit transfer attempts in an availability transfer sample and specified values of supported performance parameters (Bit Error Probability, Bit Loss Probability, Extra Bit Probability, User Information Bit Transfer Rate, and User Fraction of Input/Output Time) used in determining outcomes of such transfer samples.

Output from **TRANSF** procedures includes

- correlation results recorded in the correlator output file (**COR**),
- a summary of bit transfer failure outcomes recorded in the <u>bit transfer outcome file</u> (**B10**),
- outcomes of individual block transfer attempts recorded in the <u>block transfer outcome file</u> (**B20**),
- outcomes of individual availability transfer samples recorded in the <u>transfer sample outcome file</u> (**B30**), and
- a set of throughput transfer sample descriptors recorded in the <u>throughput sample outcome file</u> (**B40**).

Procedures for identifying bit and block transfer attempts are discussed in Section 4.2.1 and procedures for determining their outcomes are discussed in Section 4.2.2. Procedures for selecting and processing availability and throughput transfer samples are discussed in Section 4.2.3.

4.2.1 Identification of Bit and Block Transfer Attempts: Data Correlation

The user information transfer phase of a normal data communication session in a performance measurement test is outlined in Figure 19. User information input to the system at the source interface is partitioned into a sequence of <u>source blocks</u>. The transfer start time and binary contents for each source block

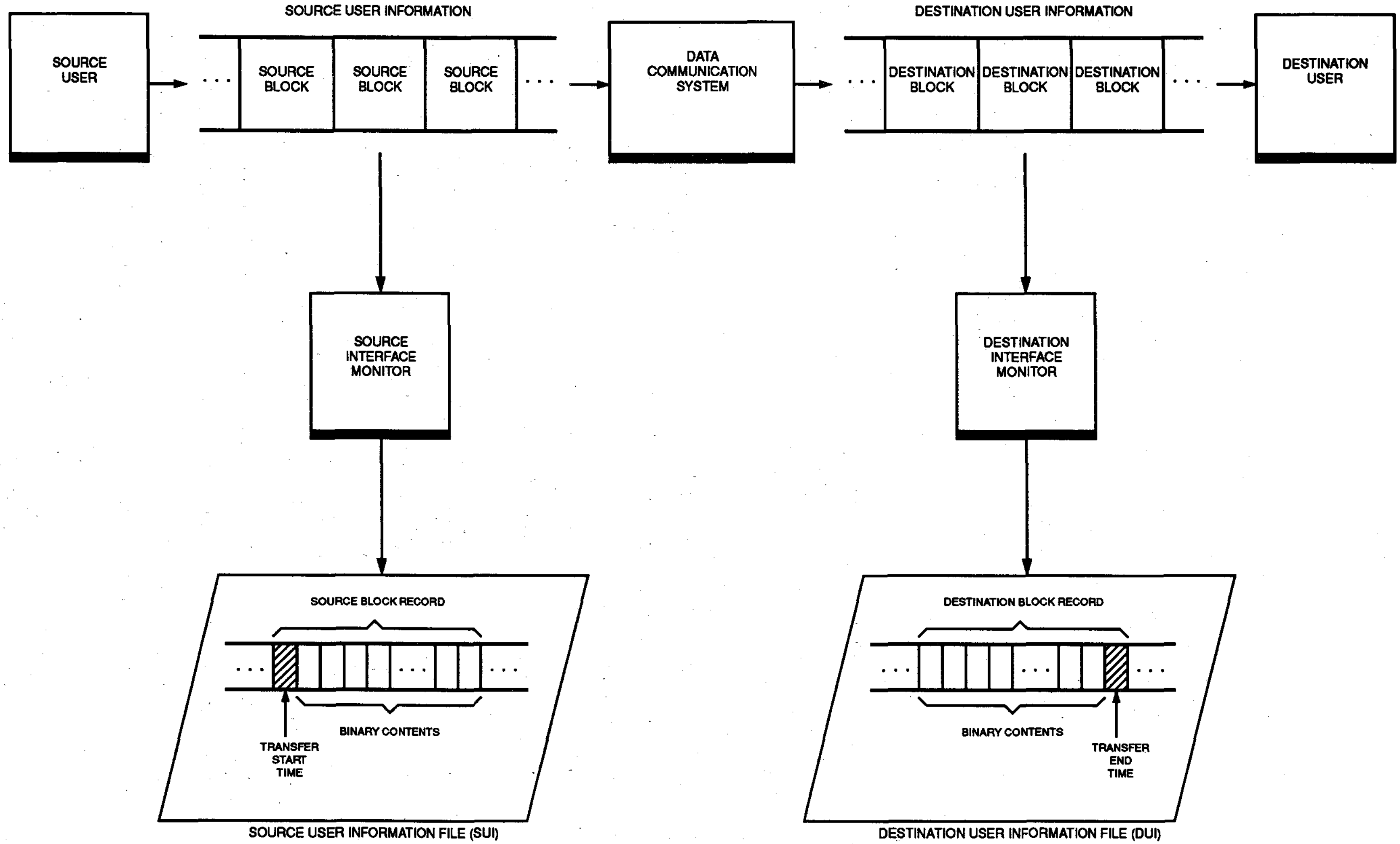


Figure 19. Outline of user information transfer phase of normal data communication session.

are recorded in the source user information file by the local interface monitor. User information output from the system at the destination interface is partitioned into a sequence of destination blocks. The transfer end time and binary contents for each destination block are recorded in the destination user information file by the local interface monitor.

The resulting records of source and destination blocks are input to the data correlation process, which is outlined in Figure 20. Subroutine **BITCOR** compares and analyzes user information in the source and destination blocks and identifies a sequence of bit transfer attempts. Subroutine **BLKCOR** partitions the sequence of bit transfer attempts identified by **BITCOR** into a sequence of block transfer attempts and identifies Start of Block Transfer and End of Block Transfer events for each attempt. Results of the data correlation process are recorded in the correlator output file (COR). Records in the correlator output file are the basis for all subsequent procedures in user information transfer performance assessment.

The discussion of data correlation presented in this section is outlined in Figure 21 and consists of three principal parts: (i) concepts and principles used in subroutine **BITCOR** to identify bit transfer attempts, (ii) concepts and principles used in subroutine **BLKCOR** to identify block transfer attempts, and (iii) contents of the correlator output file.

4.2.1.1 Identification of Bit Transfer Attempts

User information bits input to the system by the source user or output from the system to the destination user during a data communication session are categorized as shown in Figure 22. An undelivered bit is a source bit that does not correspond to any destination bit; an extra bit is a destination bit that does not correspond to any source bit. (A nonextra destination bit corresponds to a delivered source bit.) A nonextra destination bit is a correct bit if its binary value is identical to that of the corresponding source bit and is an incorrect bit otherwise (i.e., if its binary value differs from that of the corresponding source bit).

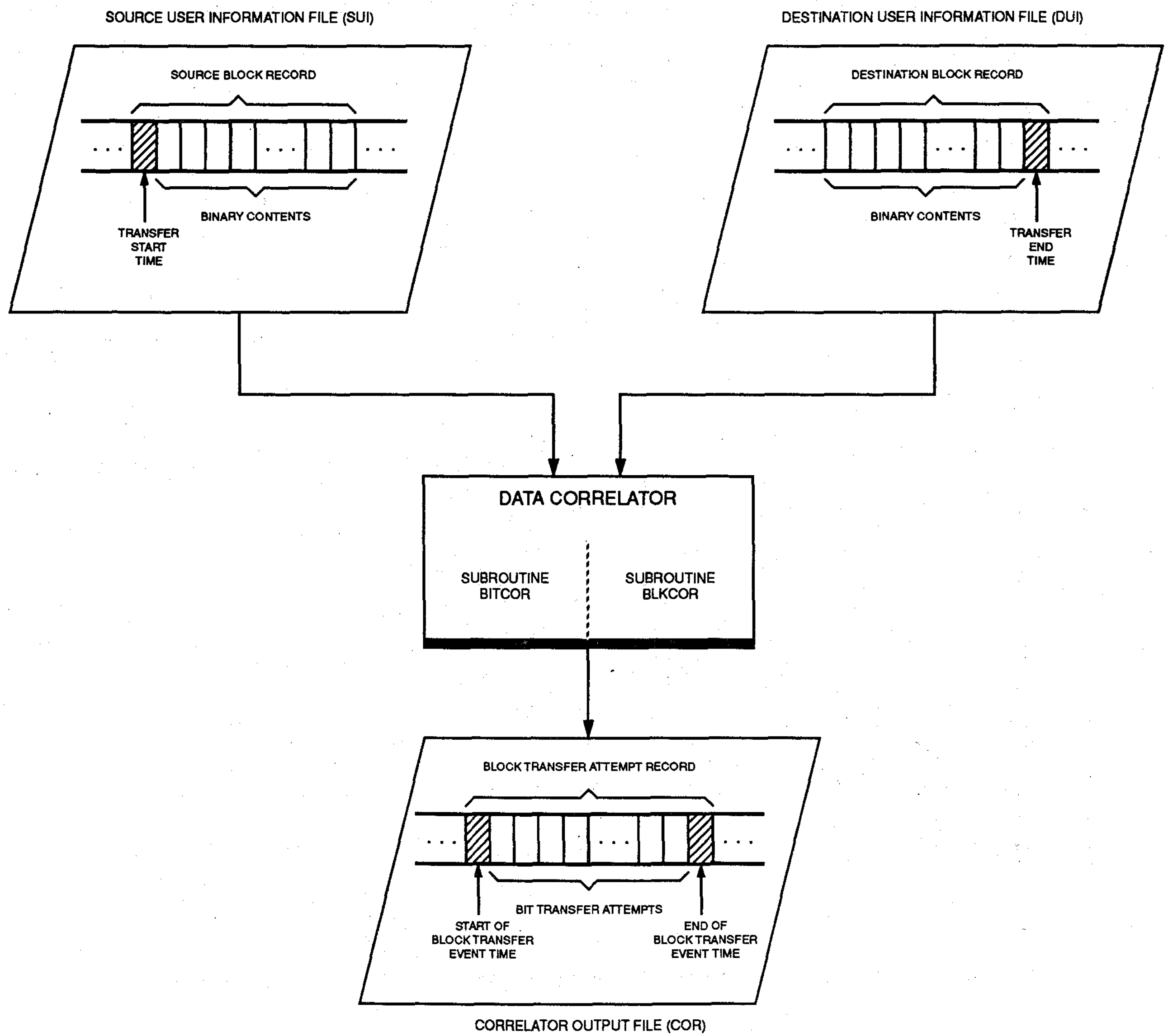


Figure 20. Outline of data correlation process.

4.2.1 Identification of Bit and Block Transfer Attempts: Data Correlation

4.2.1.1 Identification of Bit Transfer Attempts

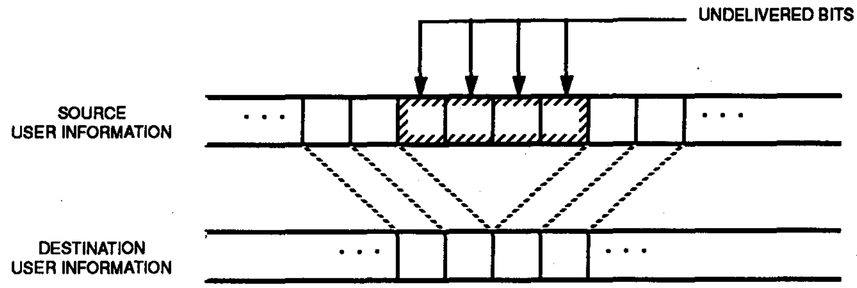
- a. Probabilistic Basis of Bit Correlation
- b. Bit Correlation Procedures
  - b.1 Identification of Clusters of Incorrect BCOs
  - b.2 Identification of Strings of Undelivered BCOs
  - b.3 Identification of Strings of Extra BCOs
  - b.4 Overall BCO Identification Process
- c. Correlation Performance
  - c.1 Absence of Bit Transfer Failures
  - c.2 Well-isolated Bit Transfer Failures
  - c.3 Poorly-isolated or Nonisolated Bit Transfer Failures

4.2.1.2 Identification of Block Transfer Attempts

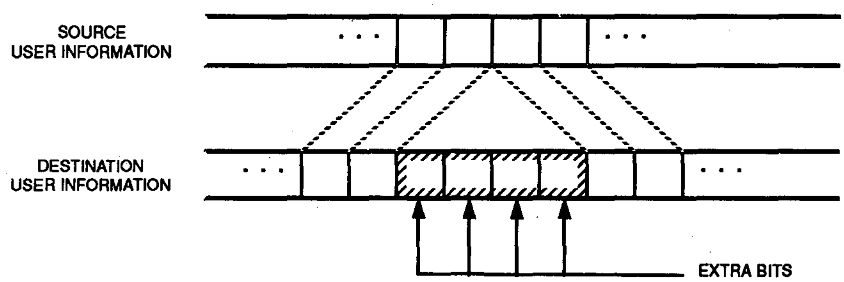
- a. BCO Assignment Algorithm
- b. Block Transfer Event Times

4.2.1.3 Correlator Output FIle

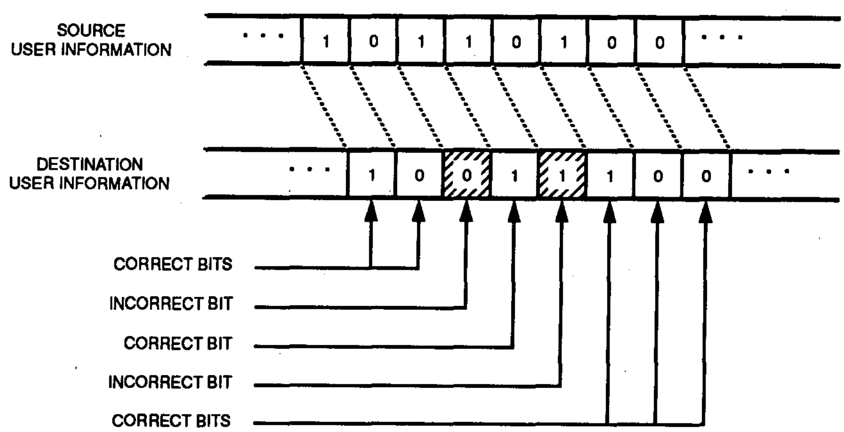
Figure 21. Organization of Section 4.2.1.



a. Undelivered Bits



b. Extra Bits



c. Correct and Incorrect Bits

Figure 22. User information bit categories.

In user information transfer performance assessment, a separate bit transfer attempt is associated with

- each pair of corresponding source and destination bits,
- each undelivered source bit, and
- each extra destination bit.

Subroutine **BITCOR** identifies each bit transfer attempt as a bit comparison outcome (BCO) in one of the following categories:

- A correct BCO is a bit transfer attempt associated with a corresponding pair of source and destination bits having the same binary value.
- An incorrect BCO is a bit transfer attempt associated with a corresponding pair of source and destination bits having different binary values.
- An undelivered BCO is a bit transfer attempt associated with an undelivered source bit.
- An extra BCO is a bit transfer attempt associated with an extra destination bit.

The BCOs associated with hypothetical sequences of source and destination bits are shown in Figure 23.

Some user information transfer failures may produce out-of-sequence destination bits.[2] Because it is generally not possible to reliably distinguish such failures (illustrated in Figure 24a) from alternative interpretations in which bit order is preserved (illustrated in Figure 24b), ANS X3.102 does not define bit transfer outcomes and parameters for bit sequence errors. Data correlation procedures in **BITCOR** assume that bit order is preserved and interpret any bit sequence errors that do occur as a combination of incorrect, undelivered, and extra bits.

---

[2]Nonextra destination bits d(n) and d(n'), where $n < n'$, are out-of-sequence bits if they respectively correspond to source bits s(m) and s(m'), where $m > m'$. In Figure 24a, d(3) and d(4) are out-of-sequence bits, whereas d(1) and d(2) are not. Bit order is said to be preserved by user information transfer in a data communication session if there are no out-of-sequence destination bits.

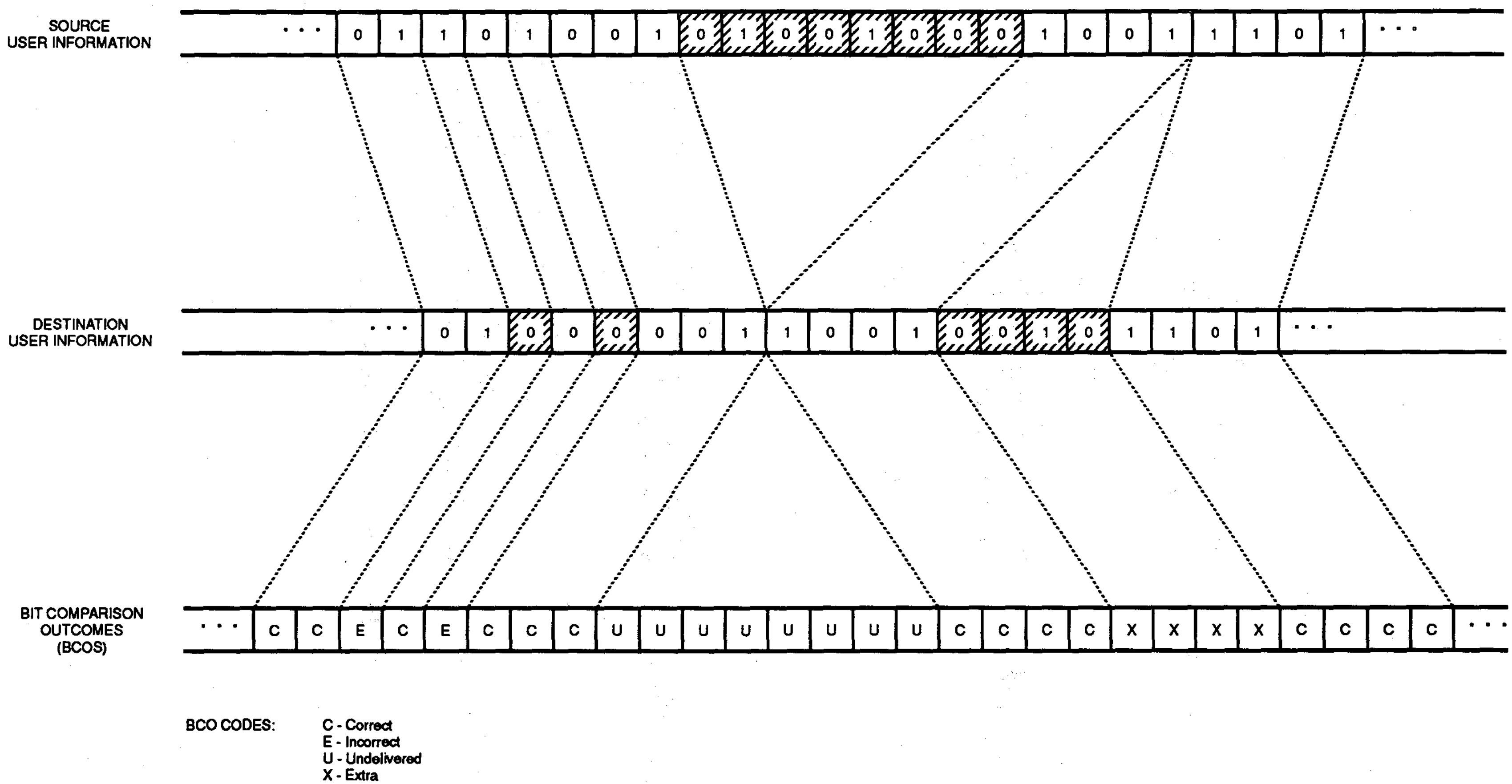
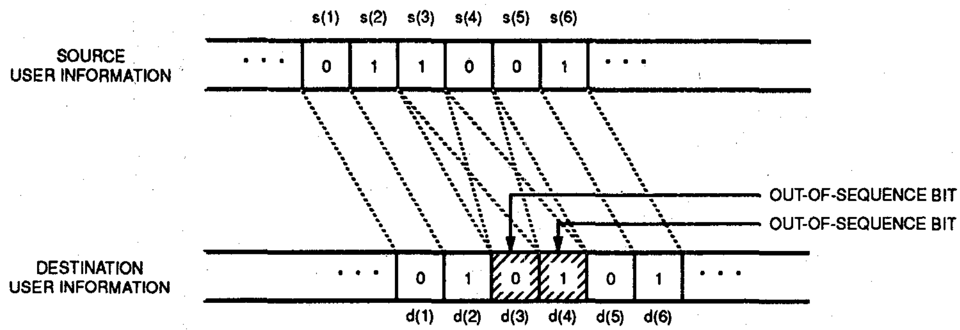
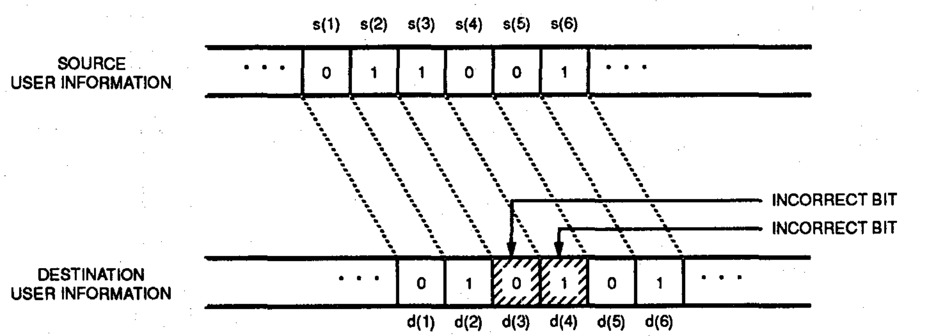


Figure 23. Bit comparison outcome (BCO) categories.



a. Out-of-Sequence Bits



b. Alternative Interpretation as Incorrect Bits

Figure 24. Out-of-sequence bits.

a. **Probabilistic Basis of Bit Correlation**

It is assumed that the occurrence of incorrect, undelivered, and extra BCOs in user information transfer can be described by a probability model. Bit transfer failures in modern data communication systems are generally infrequent and in most cases result in (i) an isolated cluster of (not necessarily adjacent) incorrect BCOs, (ii) an isolated string of undelivered BCOs, or (iii) an isolated string of extra BCOs.[3] Such behavior implies a model in which there is

- a very small probability that a randomly selected BCO is a noncorrect BCO of a given type, and
- a feature that describes the tendency of incorrect BCOs to occur in clusters and the tendency of undelivered and extra BCOs to occur in strings.[4]

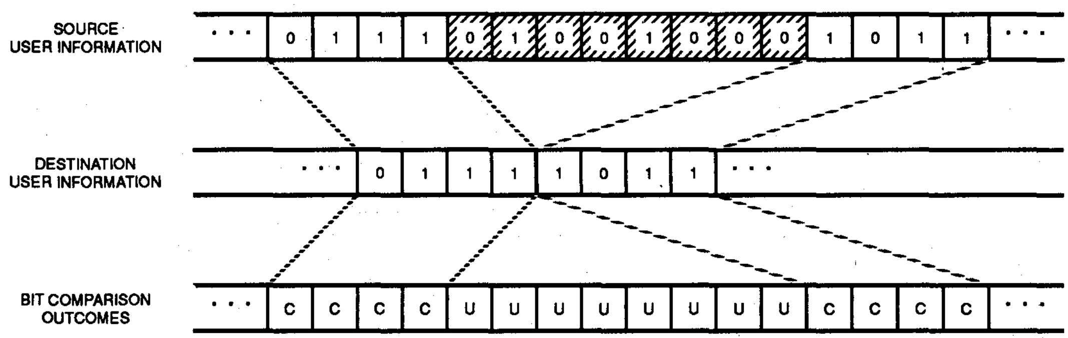
A given input sequence of source bits and the resulting output sequence of destination bits determine a set of possible configurations of corresponding bit pairs, undelivered bits, and extra bits that can account for the observed input and output bit sequences. Several such configurations and the associated BCOs are illustrated in Figure 25 for a particular pair of source and destination bit sequences.

In the general case (where incorrect, undelivered, and extra bits may occur, but out-of-sequence bits are excluded), possible configurations are constrained only by the obvious conditions that (i) the number of corresponding bit pairs plus the number of undelivered bits is equal to the number of source bits and (ii) the number of corresponding bit pairs plus the number of extra bits

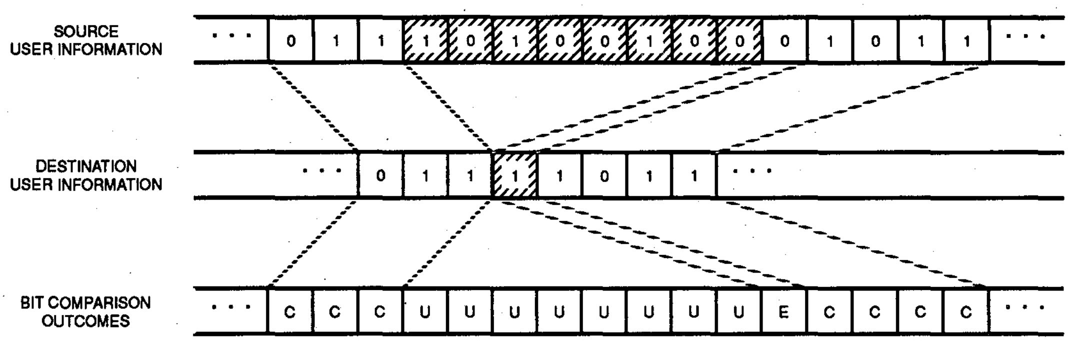
---

[3]A string of undelivered BCOs, a cluster of incorrect BCOs, or a string of extra BCOs in a BCO sequence is <u>isolated</u> if (i) it is immediately preceded in the sequence by one or more correct BCOs or is at the beginning of the sequence, and (ii) it is immediately followed in the sequence by one or more correct BCOs or is at the end of the sequence.

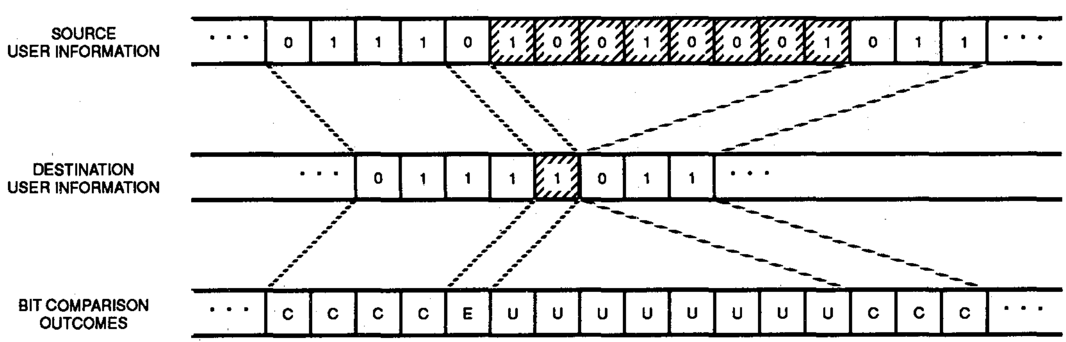
[4]An example of such a feature is a model in which successive bit transfer attempts are treated as a stationary first-order Markov process. In the Markov model, the outcome of a bit transfer attempt may be influenced by the outcome of the immediately preceding attempt, but not by the outcome of any attempt earlier than that. This model may be implemented by means of the (relatively large) conditional probability that any BCO is a noncorrect BCO of a specific type, given that the preceding BCO is of the same type. Program **STAR**, discussed in Volume 5 of this report, uses a Markov model in the analysis of performance failures.



a. Isolated String of Undelivered Bits



b. String of Undelivered Bits Followed by Incorrect Bit



c. Incorrect Bit Followed by String of Undelivered Bits

Figure 25. Configurations of corresponding bit pairs and undelivered bits associated with a pair of source and destination bit sequences.

is equal to the number of destination bits. In this case, the number N(m,n) of possible configurations of corresponding bit pairs, undelivered bits, and extra bits associated with a given source bit sequence and the resulting destination bit sequence is given by

$$N(m,n) = \sum_{k=0}^{K} \binom{m}{k} \binom{n}{k} ,$$

where m and n are the respective lengths of the source and destination bit sequences and $K = \min(m,n)$. Each term in the sum is the number of configurations that contain exactly k corresponding bit pairs and K is the largest number of corresponding bit pairs that can occur in any of the configurations. The value of N(m,n) grows rapidly as m and n increase; for typical values of m and n in performance measurements (many thousands of bits), N(m,n) is a very large number.

Configurations may be subject to additional constraints in the case of particular systems. For example, if user information loss always occurs as strings of characters, then undelivered bits occur only as strings whose lengths are integer multiples of the number of bits used to represent a single character. Such constraints reduce the number of possible configurations associated with a given pair of input and output bit sequences.

Each possible configuration of corresponding bit pairs, undelivered bits, and extra bits determines a sequence of BCOs. On the basis of the assumed probability model, each sequence q in the set Q(S,D) of possible BCO sequences associated with a given sequence S of source bits and the resulting sequence D of destination bits is characterized by a probability P(q) that the sequence represents the actual bit transfer attempts. Thus,

$$\sum_{q \in Q(S,D)} p(q) = 1.$$

Under such conditions, it is not possible to determine which of the possible sequences represents the actual bit transfer attempts. Instead, the objective of the bit correlation process is to identify the most probable sequence $\hat{q}$ of associated BCOs.

b. Bit Correlation Procedures

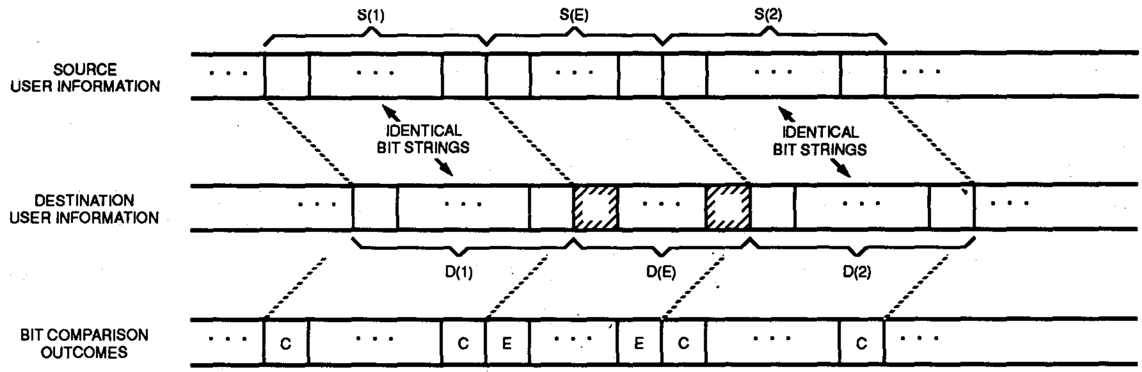
The probabilistic basis of bit correlation outlined in the preceding paragraphs provides a useful framework for describing correlation concepts, but it is seldom directly applicable in practical procedures for identifying the most probable sequence of BCOs associated with a given pair of source and destination bit sequences. In many cases, a bit transfer model that enables the evaluation of BCO sequence probabilities does not exist before data correlation is performed (correlation is often carried out in order to develop such a model). In cases where a sufficiently detailed model already exists, typical input and output bit sequences in performance measurements result in so many possible BCO sequences that it is not feasible to identify each sequence and evaluate its probability.

Subroutine BITCOR is designed to identify the most probable sequence $\hat{q}$ of associated BCOs for user information transfer measurement periods in which bit transfer failures either do not occur or result only in clusters of incorrect BCOs, strings of undelivered BCOs, or strings of extra BCOs that are well isolated by strings of correct BCOs. When all noncorrect BCO entities are well isolated, the input and output bit sequences exhibit distinctive patterns of identical source and destination strings. These patterns and the associated BCOs are shown in Figure 26 for cases that illustrate each indicated type of bit transfer failure.

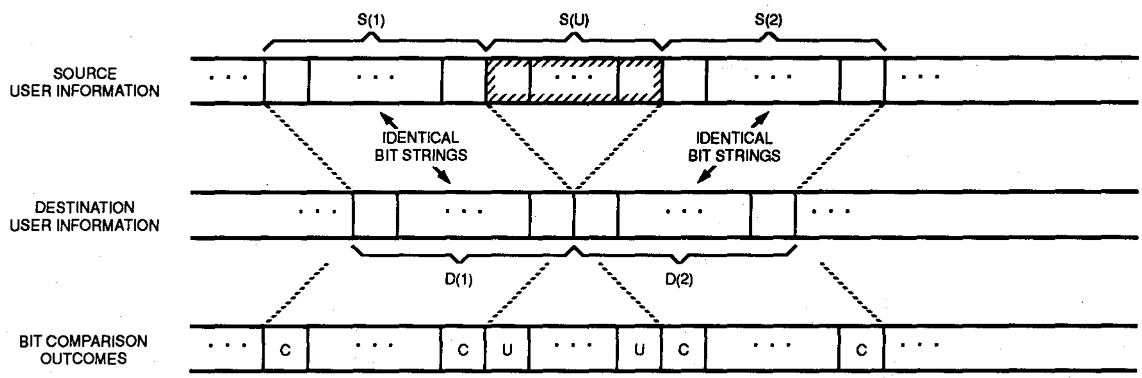
In Figure 26a, a cluster D(E) of incorrect bits is preceded by a string D(1) and followed by a string D(2), where D(1) corresponds[5] to and is identical to the source string S(1) and D(2) corresponds to and is identical to the source string S(2). The strings S(1) and S(2) are separated by a string S(E) that differs from D(E) but has the same length as D(E). Note that D(E) begins with an incorrect bit and ends with an incorrect bit; intervening bits are either correct or incorrect. The key feature of the input and output bit sequences in this case is that two nonadjacent source strings separated by a certain number of bits are respectively identical to two destination strings separated by the same number of bits.

---

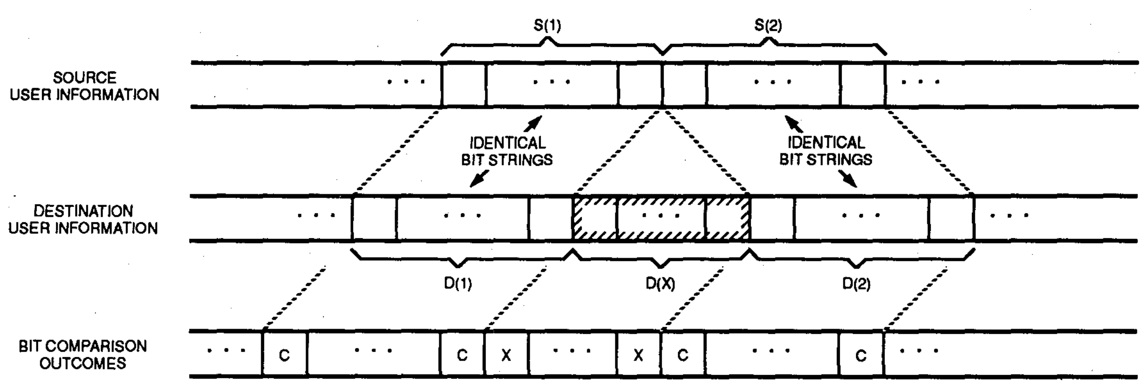
[5]A sequence S of source bits and a sequence D of destination bits <u>correspond</u> if both S and D contain the same number of bits and each bit in S corresponds to its serial counterpart in D.



a. Isolated Cluster of Incorrect Bits



b. Isolated String of Undelivered Bits



c. Isolated String of Extra Bits

Figure 26. Patterns of identical source and destination bit strings produced by isolated bit transfer failures.

In Figure 26b, a string S(U) of undelivered bits is preceded by a string S(1) and followed by a string S(2), where S(1) corresponds to and is identical to the destination string D(1) and S(2) corresponds to and is identical to the destination string D(2). The strings D(1) and D(2) are adjacent. The key feature of the input and output bit sequences in this case is that two nonadjacent source strings separated by a certain number of bits are respectively identical to two adjacent destination strings.

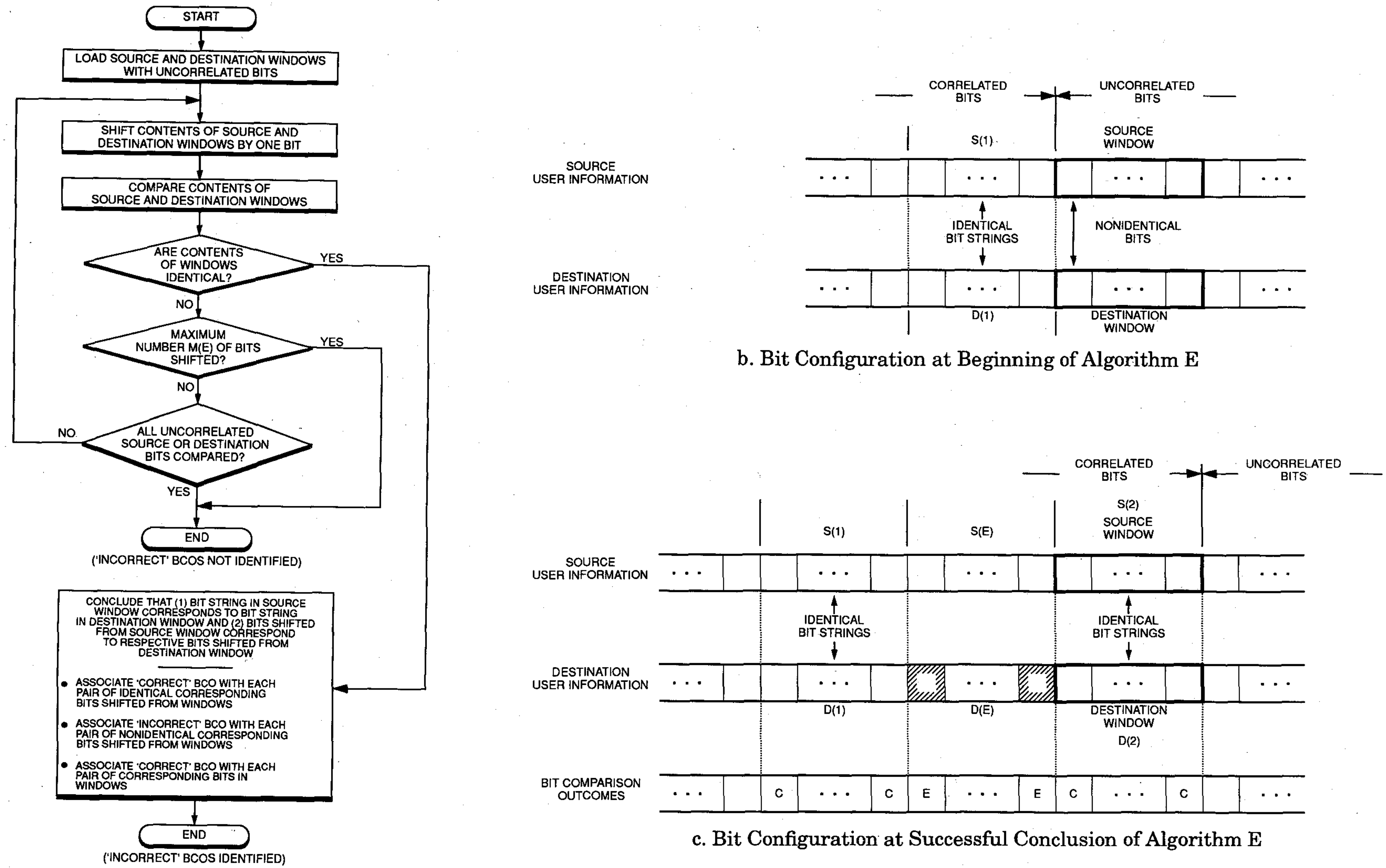
In Figure 26c, a string D(X) of extra bits is preceded by a string D(1) and followed by a string D(2), where D(1) corresponds to and is identical to the source string S(1) and D(2) corresponds to and is identical to the source string S(2). The strings S(1) and S(2) are adjacent. The key feature of the input and output bit sequences in this case is that two nonadjacent destination strings separated by a certain number of bits are respectively identical to two adjacent source strings.

In most cases where (i) bit transfer is described by the kind of model outlined in part a of this section and (ii) input and output bit sequences exhibit patterns of identical source and destination strings illustrated in Figure 26, the most probable BCO sequence has the form indicated by the figure. For example, the source and destination bit sequences in Figure 25 exhibit patterns typically produced by a well-isolated string of undelivered bits. The probability of the BCO sequence in Figure 25a, in which the only noncorrect BCO entity is a string of undelivered BCOs, is much greater than the probability of more complex configurations of noncorrect BCOs, such as those in Figures 25b and 25c.

This principle is the basis of three algorithms in BITCOR (denoted as algorithm E, algorithm U, and algorithm X) that are respectively designed to identify (i) clusters of incorrect BCOs, (ii) strings of undelivered BCOs, and (iii) strings of extra BCOs when these entities are isolated by sufficiently long strings of correct BCOs. These algorithms are described in the following paragraphs.

b.1 Identification of Clusters of Incorrect BCOs

Algorithm E used by BITCOR to identify well-isolated clusters of incorrect BCOs is outlined in Figure 27a. The algorithm is invoked only when (i) the



a. Outline of Algorithm E

Figure 27. Summary of algorithm E for identifying clusters of incorrect BCOs.

first uncorrelated[6] source bit differs from the first uncorrelated destination bit and (ii) these bits are preceded by identical source and destination strings (strings S(1) and D(1) in Figure 26a) or are the initial bits in the respective user information files. If identical source and destination strings precede the first uncorrelated bits, BITCOR has assumed that these strings correspond and has associated a correct BCO with each pair of corresponding bits.

The initial step in algorithm E loads a string beginning with the first uncorrelated source bit into one array (called the <u>source user information window</u>) and loads a string beginning with the first uncorrelated destination bit into another array (called the <u>destination user information window</u>). The starting bit configuration is shown in Figure 27b; note that the initial uncorrelated source and destination bits are not identical. Uncorrelated source and destination user information is then shifted in one-bit steps through the respective windows. Contents of the two windows are compared after each shift. The shift-compare process is continued until

- the bit string in the source window is identical to the bit string in the destination window,
- a specified maximum number M(E) of bits have been shifted from each window, or
- no uncorrelated source bits or uncorrelated destination bits remain to be compared.

If contents of the two windows are identical, BITCOR concludes that the bit string in the source window (string S(2) in Figure 26a) corresponds to the bit string in the destination window (string D(2) in Figure 26a). The routine also concludes that bits shifted from the source window (string S(E) in Figure 26a) correspond to the respective bits shifted from the destination window (string D(E) in Figure 26a). A correct BCO is associated with each identical pair of corresponding bits shifted from the windows and an incorrect BCO is associated

---

[6]With respect to the correlation process for a given performance data batch, a source or destination bit is said to be <u>correlated</u> if (i) the bit has been associated with a particular BCO and (ii) the BCO has been associated with a particular block transfer attempt. Otherwise, a source or destination bit is said to be <u>uncorrelated</u>.

with each nonidentical pair of corresponding bits shifted from the windows. A correct BCO is associated with each pair of corresponding bits in the windows. The ending bit configuration and the associated BCOs are shown in Figure 27c.

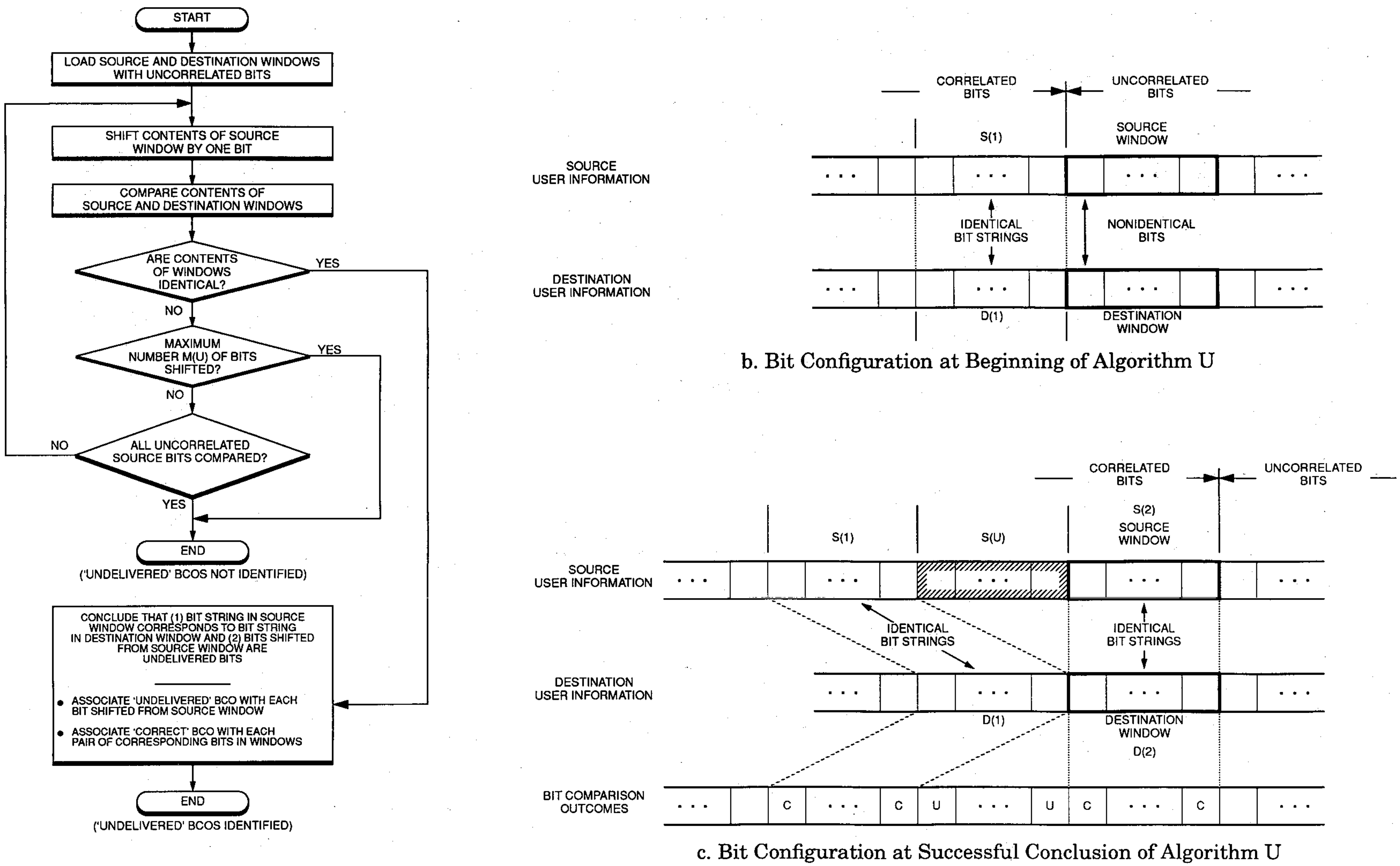
b.2 Identification of Strings of Undelivered BCOs

Algorithm U used by **BITCOR** to identify well-isolated strings of undelivered BCOs is outlined in Figure 28a. Starting conditions and the initialization of user information windows with uncorrelated bits are the same as those described earlier for algorithm E. Thus, algorithm U is invoked only when (i) the first uncorrelated source bit differs from the first uncorrelated destination bit and (ii) these bits are preceded by identical source and destination strings (strings S(1) and D(1) in Figure 26b) or are the initial bits in the respective user information files. If identical source and destination strings precede the first uncorrelated bits, **BITCOR** has assumed that these strings correspond and has associated a correct BCO with each pair of corresponding bits.

The initial step in algorithm U loads a string beginning with the first uncorrelated source bit into the source user information window and loads a string beginning with the first uncorrelated destination bit into the destination user information window. The starting bit configuration is shown in Figure 28b. Uncorrelated user information is then shifted in one-bit steps through the source window while contents of the destination window remain fixed. Contents of the two windows are compared after each shift. The shift-compare process is continued until

- the bit string in the source window is identical to the bit string in the destination window,
- a specified maximum number M(U) of bits have been shifted from the source window, or
- no uncorrelated source bits remain to be compared.

If contents of the two windows are identical, **BITCOR** concludes that the bit string in the source window (string S(2) in Figure 26b) corresponds to the bit string in the destination window (string D(2) in Figure 26b) and that all bits shifted from the source window (string S(U) in Figure 26b) are undelivered bits. An undelivered BCO is thus associated with each bit shifted from the source



a. Outline of Algorithm U

Figure 28. Summary of algorithm U for identifying strings of undelivered BCOs.

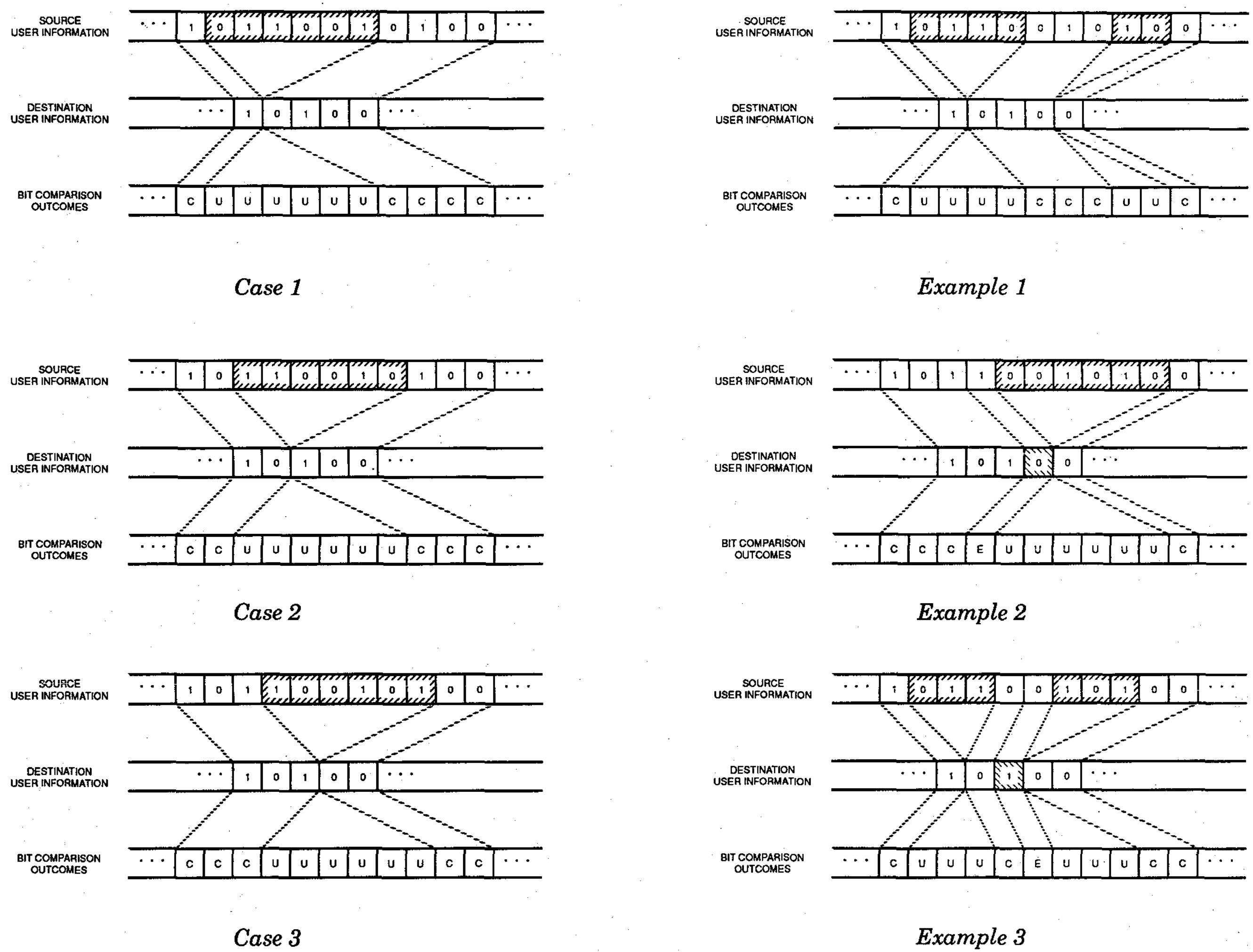
window and a correct BCO is associated with each pair of corresponding bits in the windows. The ending bit configuration is shown in Figure 28c.

In the example shown in Figure 25a, there is a single sequence $\hat{q}$ of associated BCOs whose probability substantially exceeds that of every other associated BCO sequence. When bit transfer failures result in a well-isolated string of undelivered bits, there often exists a set $\hat{Q}$ consisting of two or more sequences of associated BCOs such that (i) all sequences in $\hat{Q}$ have the same probability and (ii) the probability of a sequence in $\hat{Q}$ substantially exceeds the probability of any associated BCO sequence not in $\hat{Q}$. An example of a set $\hat{Q}$ that contains three such BCO sequences is shown in Figure 29a. Each sequence in $\hat{Q}$ includes a string of six undelivered BCOs and represents a case where (i) one or more bits at the beginning of an undelivered string are identical to bits that immediately follow the string or (ii) one or more bits at the end of an undelivered string are identical to bits that immediately precede the string. The sequences in $\hat{Q}$ differ only in the location of the indicated strings of undelivered BCOs within the overall sequence of BCOs. All sequences of associated BCOs not in $\hat{Q}$, three of which are shown in Figure 29b, contain more complex configurations of noncorrect BCOs and have probabilities that are substantially less than the probability of a BCO sequence in $\hat{Q}$.

In cases such as that illustrated in Figure 29, algorithm U identifies the BCO sequence that has the longest string of correct BCOs preceding the undelivered BCOs (e.g., the last BCO sequence in Figure 29a). Instances in which block transfer outcomes are affected by such a choice are described in Section 4.2.2.3.

b.3 Identification of Strings of Extra BCOs

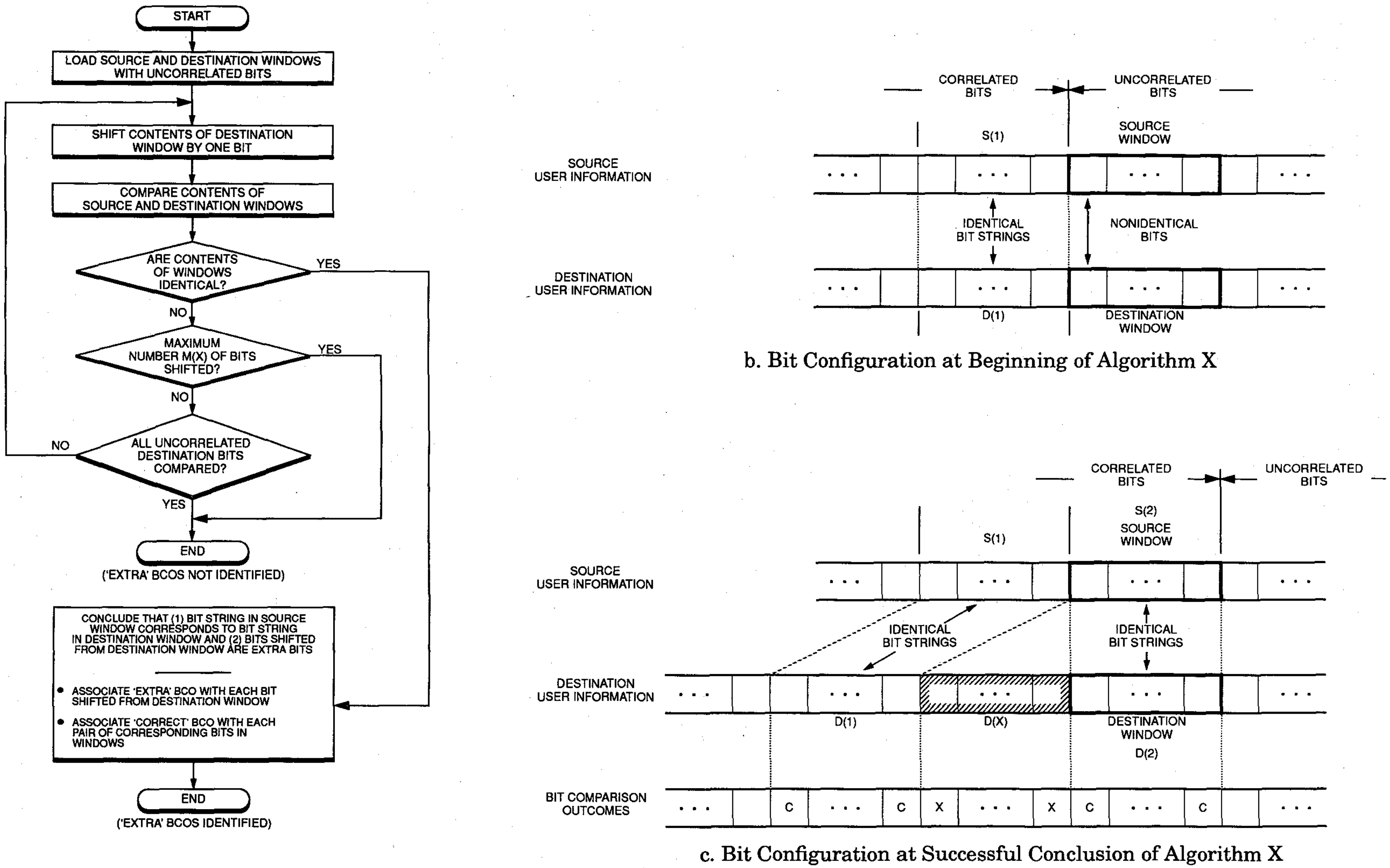
Algorithm X used by **BITCOR** to identify well-isolated strings of extra BCOs is outlined in Figure 30a. This algorithm is equivalent to interchanging source and destination roles in algorithm U for identifying undelivered BCOs. Starting conditions and the initialization of user information windows with uncorrelated bits are the same as those described earlier for algorithm E and algorithm U. The starting bit configuration is shown in Figure 30b. Uncorrelated user information is then shifted in one-bit steps through the destination window while contents of the source window remain fixed. Contents of the two windows are compared after each shift. The shift-compare process is continued until



a. Most Probable BCO Sequences

b. Examples of Less Probable BCO Sequences

Figure 29. Typical multi-sequence set $\hat{Q}$ of most probable BCO sequences.



b. Bit Configuration at Beginning of Algorithm X

c. Bit Configuration at Successful Conclusion of Algorithm X

a. Outline of Algorithm X

Figure 30. Summary of algorithm X for identifying strings of extra BCOs.

- the bit string in the source window is identical to the bit string in the destination window,
- a specified maximum number M(X) of bits have been shifted from the destination window, or
- no uncorrelated destination bits remain to be compared.

If contents of the two windows are identical, **BITCOR** concludes that the bit string in the source window (string S(2) in Figure 26c) corresponds to the bit string in the destination window (string D(2) in Figure 26c) and that all bits shifted from the destination window (string D(X) in Figure 26c) are extra bits. An extra BCO is thus associated with each bit shifted from the destination window BCO and a correct BCO is associated with each pair of corresponding bits in the windows. The ending bit configuration is shown in Figure 30c.

When bit transfer failures result in a well-isolated string of extra bits, there are often two or more sequences of associated BCOs in the set $\hat{Q}$ of most probable BCO sequences. The situation is analogous to that described for undelivered bits in part b.2 of this section. In such cases, algorithm X identifies the BCO sequence that has the longest string of correct BCOs preceding the extra BCOs. Instances in which block transfer outcomes are affected by such a choice are described in Section 4.2.2.3.

b.4 Overall BCO Identification Process

The overall BCO identification process used in **BITCOR** is outlined by the flowchart in Figure 31. At the start of the process or after a correlation algorithm has been successfully completed (i.e., after an algorithm has identified and processed a sequence of BCOs), a <u>selection scheme</u> (S) branches to the appropriate correlation procedure.

If the selection scheme determines that (i) uncorrelated user information includes both source and destination bits and (ii) uncorrelated source user information and uncorrelated destination user information begin at field (word) boundaries in the respective user information records (these data structures are described in Section 3.2.3), the <u>fast correlation algorithm</u> (F) compares the first uncorrelated source field and the first uncorrelated destination field. If the compared fields contain identical bit strings, **BITCOR** assumes that the two

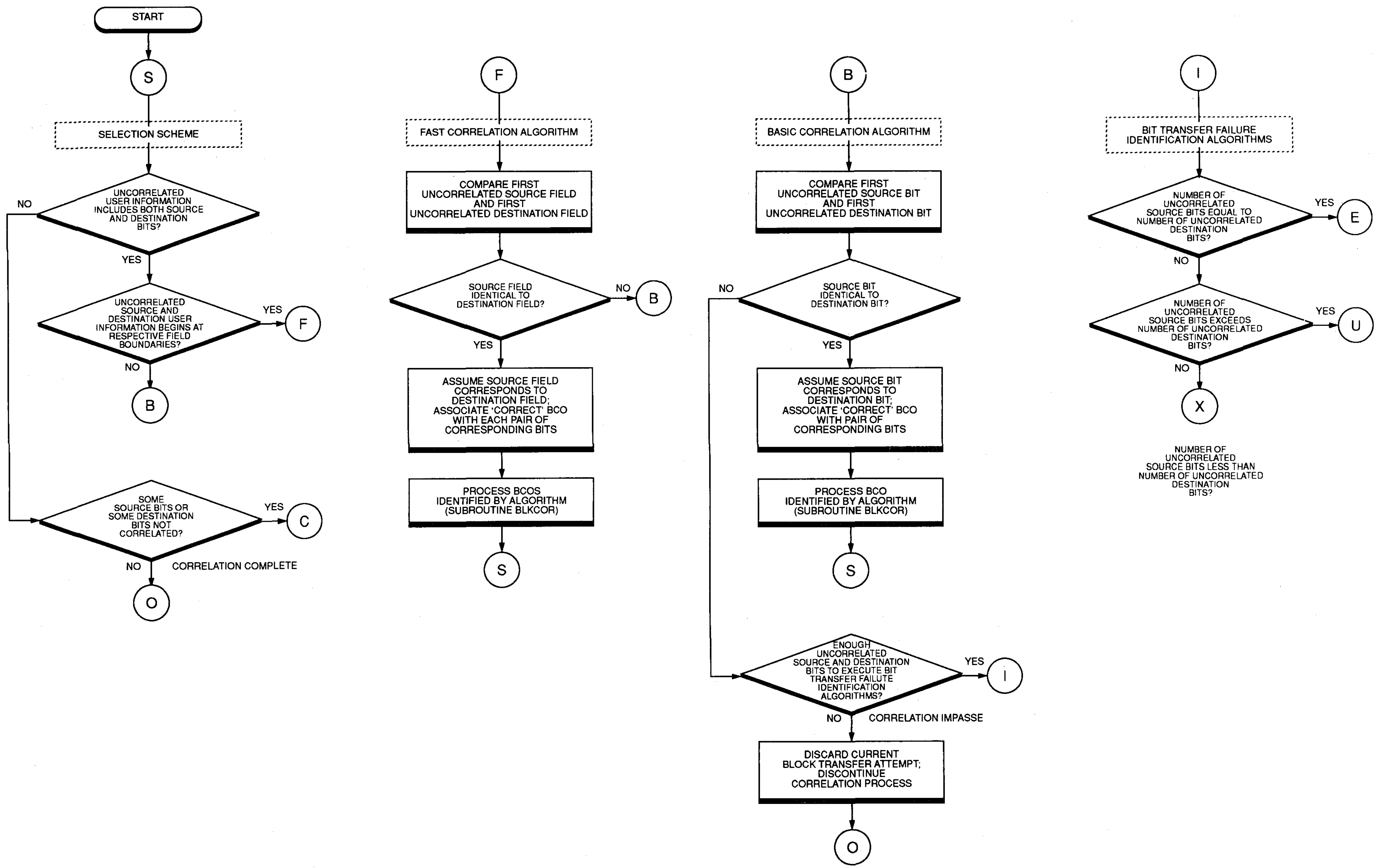


Figure 31. Outline of overall BCO identification process used by **BITCOR**.

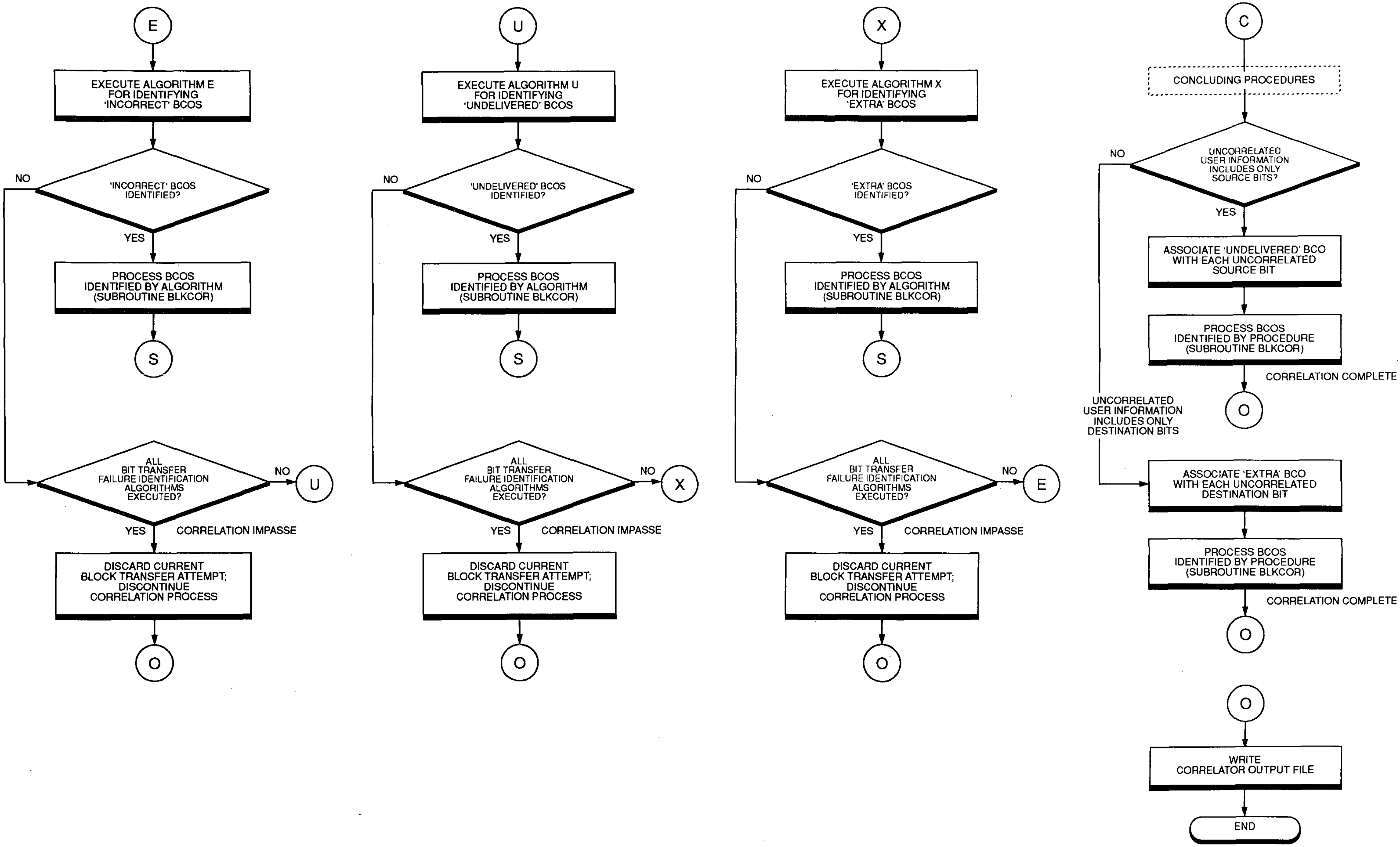


Figure 31. Outline of overall BCO identification process used by **BITCOR** (Continued).

strings correspond and associates a correct BCO with each corresponding pair of source and destination bits.

If the user information fields compared by the fast correlation algorithm do not contain identical bit strings, or if the selection scheme determines that (i) uncorrelated user information includes both source and destination bits and (ii) uncorrelated source user information or uncorrelated destination user information does not begin at a field boundary in the respective user information record, the basic correlation algorithm (B) compares the first uncorrelated source bit and the first uncorrelated destination bit. If the compared bits are identical, **BITCOR** assumes that the two bits correspond and associates a correct BCO with the pair.

If (i) the user information bits compared by the basic correlation algorithm are not identical and (ii) numbers of uncorrelated source and destination bits are sufficiently large, a **BITCOR** procedure (I) attempts to identify a cluster of incorrect BCOs, a string of undelivered BCOs, or a string of extra BCOs by using the bit transfer failure identification algorithms described previously. The search for any particular bit transfer failure begins with the algorithm deemed most likely to succeed: the incorrect BCO identification algorithm (E), the undelivered BCO identification algorithm (U), or the extra BCO identification algorithm (X) when the number of uncorrelated source bits is respectively equal to, greater than, or less than the number of uncorrelated destination bits. The search continues until some algorithm succeeds or all algorithms fail. If no bit transfer failure identification algorithm is successful, a correlation impasse is said to occur and the correlation process is discontinued after discarding the current block transfer attempt.

If the selection scheme determines that uncorrelated user information includes only source bits or only destination bits, the correlation process is completed by concluding correlation procedures (C). These procedures associate an undelivered BCO with each uncorrelated source bit or associate an extra BCO with each uncorrelated destination bit. When the correlation process is complete or a correlation impasse occurs, an output procedure (O) writes correlation results to the correlator output file as described in Section 4.2.1.3.

c. Correlation Performance

The objective of the bit correlation process, as stated earlier in this section, is to identify the most probable sequence of BCOs associated with the source and destination user information in a given data batch. Results produced by correlation procedures in subroutine **BITCOR** can be grouped into performance categories analogous to those used to classify outcomes of data communication functions. Correlation performance categories are defined as follows:

- Successful Performance. All user information bits in the given batch are correlated, and the BCO sequence identified by **BITCOR** is the most probable sequence of BCOs associated with the given data.

- Incorrect Performance. All user information bits in the given batch are correlated, but the BCO sequence identified by **BITCOR** differs from the most probable sequence of BCOs associated with the given data.

- Nonperformance. Not all user information bits in the given batch are correlated.

The performance of correlation procedures in **BITCOR** depends on (i) the distribution of bit transfer failures in the recorded user information, (ii) specified values of the user information window size L(w) and maximum bit shifts M(E), M(U), and M(X) in the respective algorithms E, U, and X for identifying well-isolated noncorrect BCO entities, and (iii) attributes of the source user information (e.g., autocorrelation). The paragraphs that follow discuss **BITCOR** correlation performance when the user information transfer measurement period includes (i) no bit transfer failures, (ii) only well-isolated bit transfer failures, and (iii) some poorly-isolated or nonisolated bit transfer failures.

c.1 Absence of Bit Transfer Failures

When no bit transfer failures occur during a user information transfer measurement period, the sequence of all source bits corresponds to and is identical to the sequence of all destination bits. Whenever the sequence of all source bits is identical to the resulting sequence of all destination bits, the most probable sequence $\hat{q}$ of associated BCOs is obtained by assuming that the source and destination bit sequences correspond; this BCO sequence consists of correct BCOs. As indicated in part b.4 of this section, the correlation procedures in **BITCOR** identify $\hat{q}$, so these procedures result in successful performance.

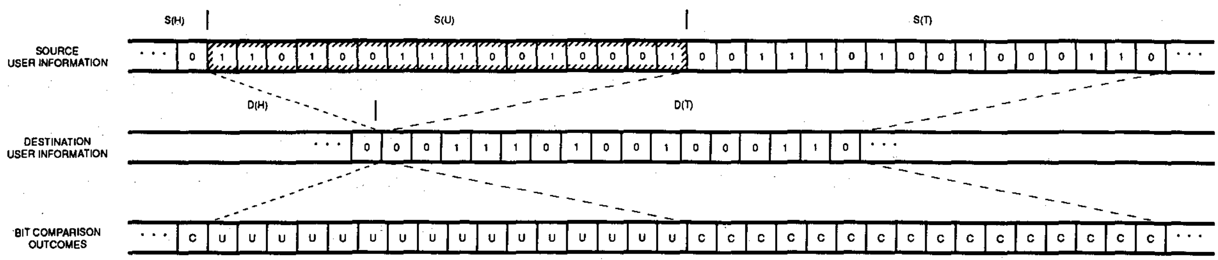
c.2 Well-isolated Bit Transfer Failures

When bit transfer failures during a user information transfer measurement period result in a sequence of associated BCOs that consists of one or more noncorrect BCO entities that are well isolated by strings of correct BCOs, this BCO sequence is usually identical to the most probable sequence $\hat{q}$ of associated BCOs. **BITCOR** correlation procedures identify $\hat{q}$ (and result in successful performance) if the applicable algorithms identify the postulated noncorrect BCO entities. From descriptions of algorithm E, algorithm U, and algorithm X presented earlier in this section, these BCO entities can be identified only if

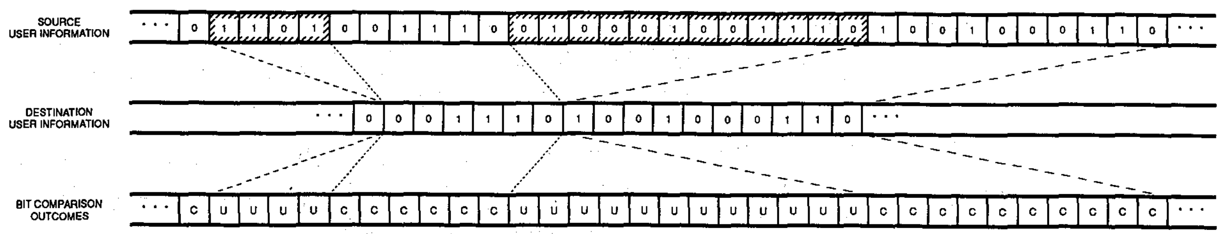
- the length of each noncorrect BCO entity in $\hat{q}$ is equal to or less than the specified maximum bit shift M(E), M(U), or M(X) in the applicable identification algorithm, and
- each noncorrect BCO entity in $\hat{q}$ is followed by a string of correct BCOs whose length is equal to or greater than the specified user information window size L(w).

The first condition above is certainly satisfied if the specified maximum bit shift in each BCO identification algorithm is the larger of the number of source user information bits and the number of destination user information bits. Except in rare cases of prolonged impairments, much smaller values of these maximum bit shifts--e.g., the equivalent of several source blocks--are usually sufficient. The smaller values may avoid excessive execution time during which a BCO identification algorithm unnecessarily persists in a futile search for a particular type of impairment when the actual impairment is of another type.

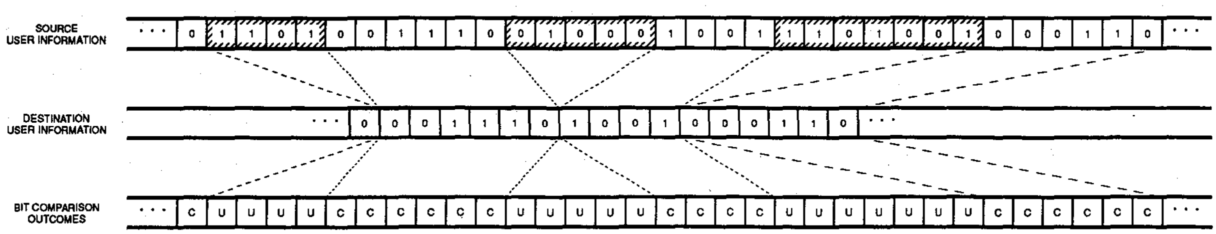
A suitable value for the user information window size L(w) cannot be so easily specified. If L(w) exceeds the length of the shortest string of correct BCOs that follow a noncorrect BCO entity in $\hat{q}$, **BITCOR** correlation procedures usually result in nonperformance (a correlation impasse). On the other hand, if L(w) is excessively small, **BITCOR** correlation procedures are apt to result in incorrect performance, as illustrated by the following hypothetical (but typical) example. In Figure 32a, the head S(H) and tail S(T) of a sequence of source bits are respectively identical to the head D(H) and tail D(T) of the resulting sequence of destination bits. The strings D(H) and D(T) are adjacent, whereas



a. Most Probable BCO Sequence, $\hat{q}$



b. BCO Sequence Identified by BITCOR when L(w)=5 or L(w)=6



c. BCO Sequence Identified by BITCOR when L(w)=4

Figure 32. Effect of user information window size on BITCOR correlation performance.

the strings S(H) and S(T) are separated by a string S(U) of 16 undelivered bits. The pattern of identical source and destination bit strings is a particular case of that shown in Figure 26b. The most probable sequence $\hat{q}$ of BCOs associated with the source and destination bit sequences in Figure 32a is identical to the postulated BCO sequence and is indicated in the figure. The sequence $\hat{q}$ is identified by correlation procedures in **BITCOR** when the specified maximum bit shift M(U) in algorithm U is equal to or greater than 16, and the user information window size L(w) is greater than 6 and equal to or less than the length of the tail S(T).

Two less probable sequences of BCOs associated with the same hypothetical source and destination bit sequences are also shown in Figure 32. The sequence in Figure 32b is identified by **BITCOR** correlation procedures when L(w) = 5 or L(w) = 6, and the sequence in Figure 32c is identified when L(w) = 4. Any of these values of L(w) result in incorrect performance.

Each of the **BITCOR** algorithms designed to identify a well-isolated noncorrect BCO entity of a particular type systematically selects and compares strings of uncorrelated source and destination bits as described earlier in this section (in parts b.1, b.2, and b.3); the strings being compared are stored in the respective user information windows. If the contents of the two windows are identical, **BITCOR** concludes the bit string in the source window corresponds to the bit string in the destination window. However, this conclusion is always subject to uncertainty - it is possible that some (or all) bit pairs in the windows do not correspond and are only fortuitously identical. Such fortuitous identity usually results in incorrect performance (as illustrated in Figures 32b and 32c) or in nonperformance.

The presence of many identical bit strings of length L(w) in the source user information increases the chance that **BITCOR** algorithms will observe fortuitously identical source and destination bit strings during the correlation process. Thus, correlation of user information is inherently unreliable when L(w) is small. Very large values of L(w) are required for successful correlation when, for example, the source user information includes long runs of zeros or ones, or consists of ordinary text that contains frequently used words. To facilitate the correlation process and enhance the likelihood of successful correlation performance, it is best to

- employ source user information that consists of a pseudorandom sequence of bits or a pseudorandom sequence of characters drawn from a large set, and
- select the user information window size L(w) in accordance with the scheme presented later in this section.

Most cases of the type illustrated in Figure 32, where the only noncorrect BCO entities in $\hat{q}$ are well isolated and the source user information consists of a pseudorandom sequence of bits or a pseudorandom sequence of characters drawn from a large set, can be described as follows. Let $r(\hat{q})$ denote the length of the shortest string of correct BCOs that follows a noncorrect BCO entity in the most probable BCO sequence $\hat{q}$ associated with a given sequence S of source bits and the resulting sequence D of destination bits. Let Q'(S,D) be the set of all BCO sequences (associated with S and D) whose probabilities are less than that of $\hat{q}$. For a given BCO sequence q' in Q'(S,D), let $r(q') = 0$ if some noncorrect BCO entity in q' is adjacent to a subsequent noncorrect BCO entity; otherwise, if every noncorrect BCO entity in q' is followed by a string of correct BCOs, let r(q') denote the length of the shortest such string of correct BCOs. Finally, let $R' = \max\{r(q')\}$, where q' ranges over all sequences in Q'(S,D). Under the postulated conditions, R' is usually substantially smaller than $r(\hat{q})$.

Typical performance of **BITCOR** correlation procedures as a function of user information window size is summarized in Figure 33; it is assumed that the specified maximum bit shifts in all BCO identification algorithms are sufficiently large. When $L(w) \le R'$, **BITCOR** correlation procedures commonly result in incorrect performance - i.e., they identify some BCO sequence q' in the set Q'(S,D) of less probable BCO sequences, as illustrated by the examples presented in Figures 32b and 32c. When $R' < L(w) \le r(\hat{q})$, **BITCOR** correlation procedures usually result in successful performance - i.e., they identify the most probable sequence $\hat{q}$ of associated BCOs.

When $L(w) > r(\hat{q})$, algorithm E, algorithm U, and algorithm X all fail at some point where a tail $T(\hat{q})$ of $\hat{q}$ is unidentified. At this point the correlation process is discontinued (a correlation impasse occurs); **BITCOR** correlation procedures then result in nonperformance.

When all noncorrect BCO entities in $\hat{q}$ are well isolated and source user information consists of a pseudorandom sequence of bits or a pseudorandom sequence of characters drawn from a large set, it is usually possible to find a

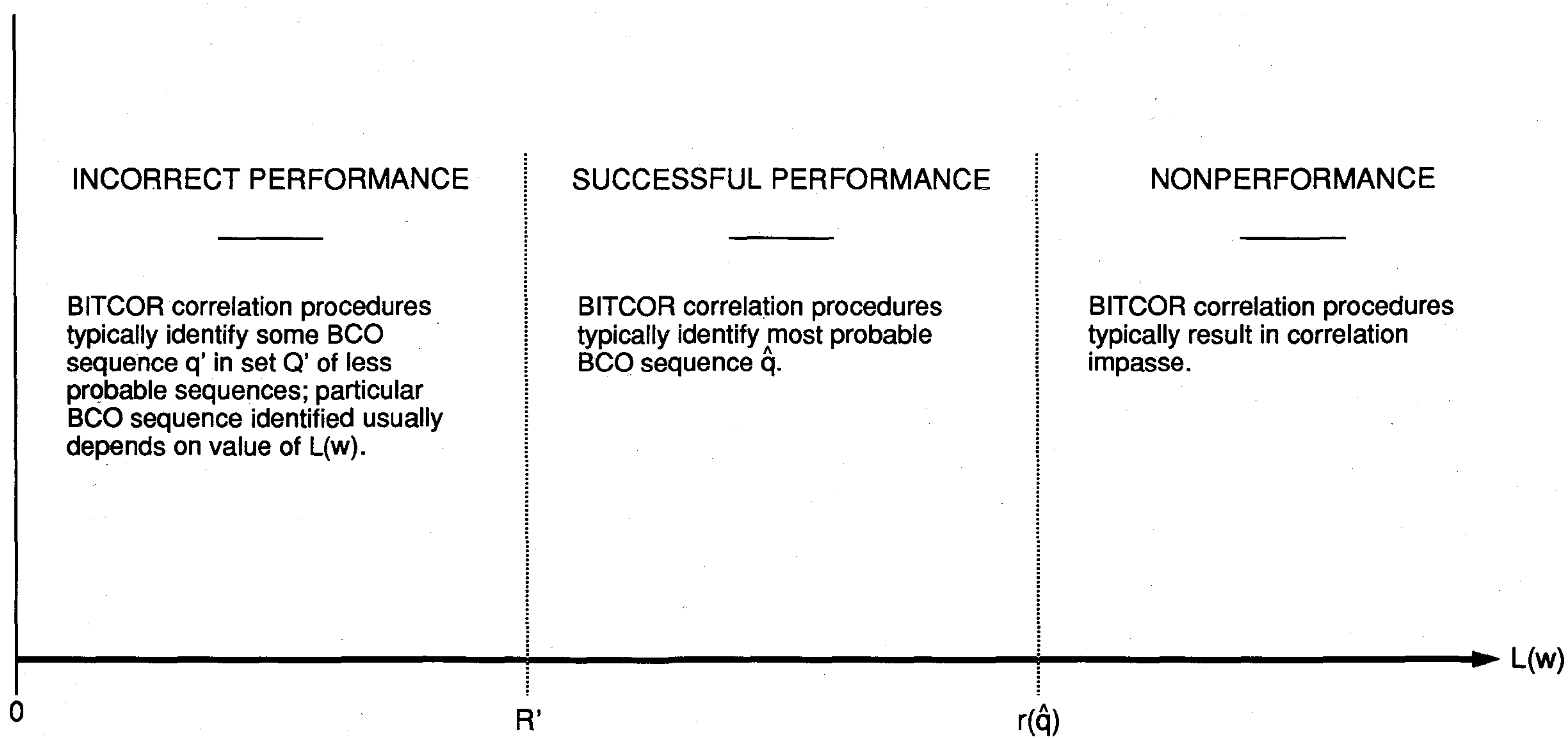


Figure 33. User information window size and BITCOR correlation performance.

user information window size L(w) such that **BITCOR** correlation procedures result in successful performance. A practical scheme for selecting an appropriate value of L(w) under these conditions is outlined in Figure 34.

Source and destination user information is first correlated with L(w) = 16. If a correlation impasse occurs, conclude that some bit transfer failure is not well isolated; such cases are discussed in part c.3 of this section.

If, in the BCO sequence q obtained by **BITCOR** when the user information window size is L(w), each noncorrect BCO entity is followed by a string of correct BCOs whose length is equal to or greater than L(w) + 16, then the same BCO sequence q will be obtained when the user information window size is L(w) + 16. Apart from rare exceptions, this condition is sufficient to conclude that the sequence q obtained by BITCOR is the most probable sequence $\hat{q}$.

When some noncorrect BCO entity in q is followed by a string of correct BCOs whose length is less than L(w) + 16, repeat the correlation process with a user information window size of L(w) + 16. If the new (larger) window size results in a correlation impasse, conclude that L(w) + 16 > $r(\hat{q})$ and that the previous correlation resulted in successful performance.

ITS has conducted extensive measurements in which source user information consisted of a pseudorandom sequence of binary bits or a pseudorandom sequence of ASCII characters drawn from a 64-character set (Wortendyke et al., 1982; Spies et al., 1988). In these measurements, a user information window size of 16 bits resulted in successful correlation performance in all but a small number of cases, where larger windows were required.

c.3 Poorly-isolated or Nonisolated Bit Transfer Failures

Bit transfer failures during a user information transfer measurement period may sometimes result in a sequence of associated BCOs that includes at least one noncorrect BCO entity separated from a subsequent noncorrect BCO entity by only a few correct BCOs. A simple example of such a BCO sequence is illustrated in Figure 35a, where a single incorrect BCO is separated from a subsequent string of undelivered BCOs by two correct BCOs. All BCOs preceding the incorrect BCO and all BCOs following the undelivered BCOs are correct BCOs. The most probable BCO sequence $\hat{q}$ in this case is identical to the postulated BCO sequence shown in the figure.

Correlation procedures in **BITCOR** result in a correlation impasse

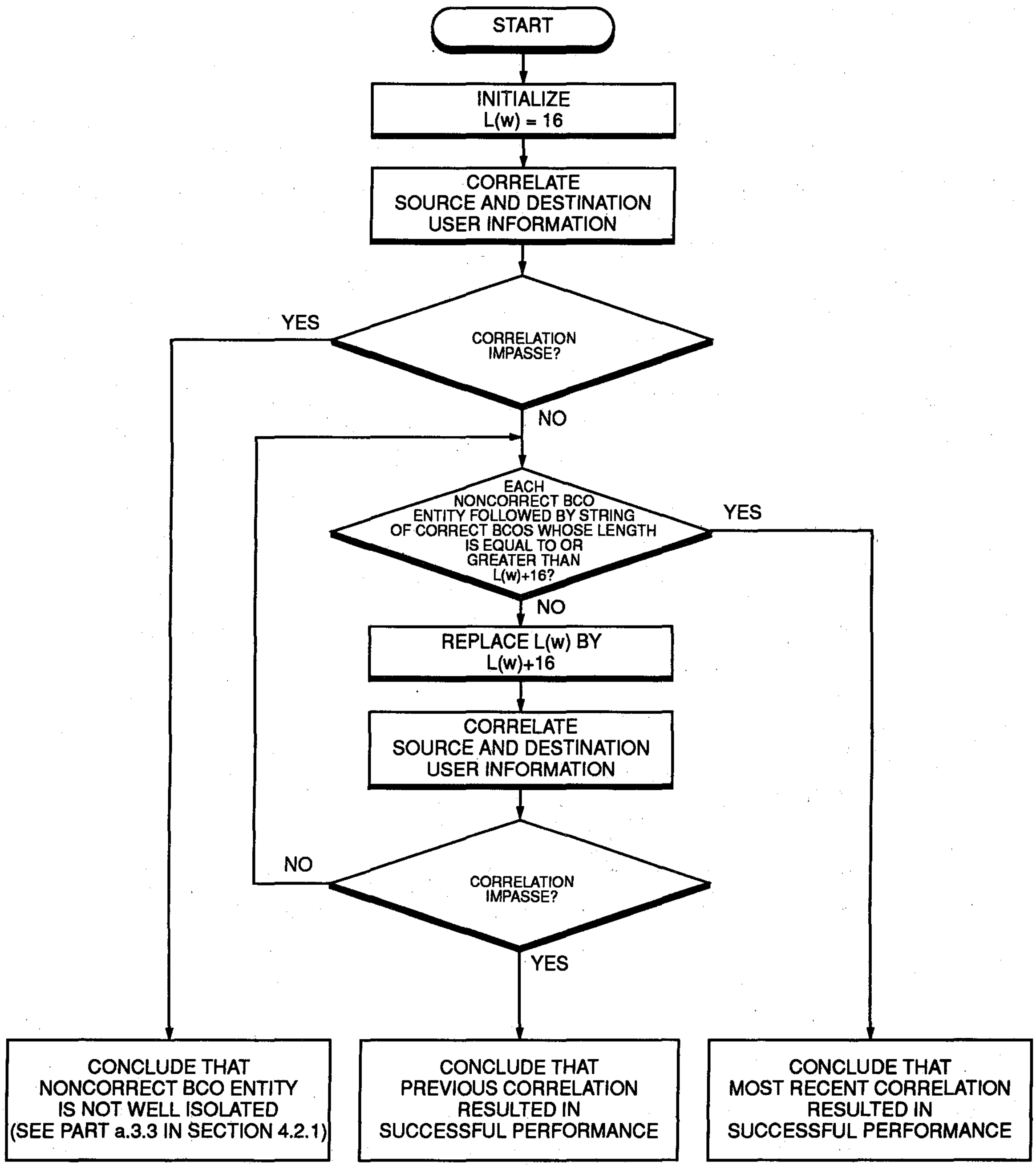
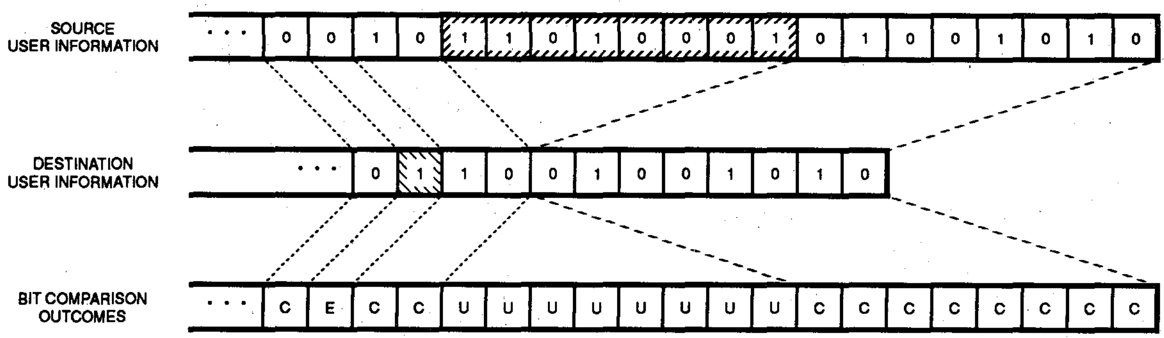
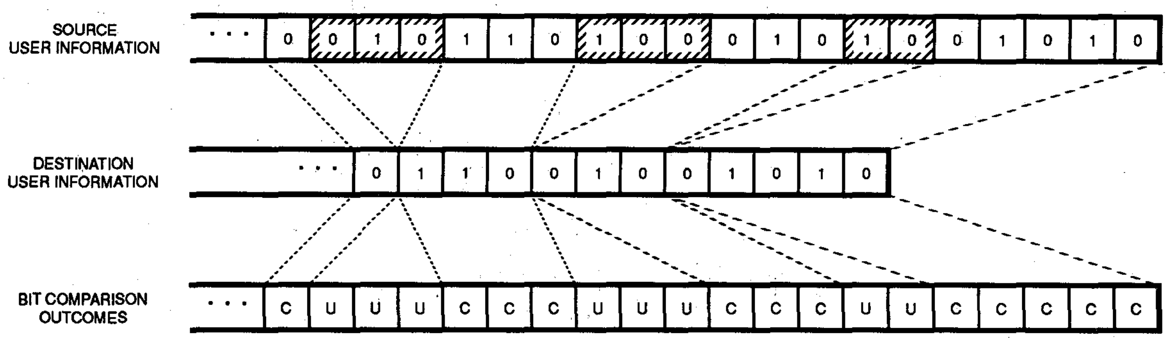


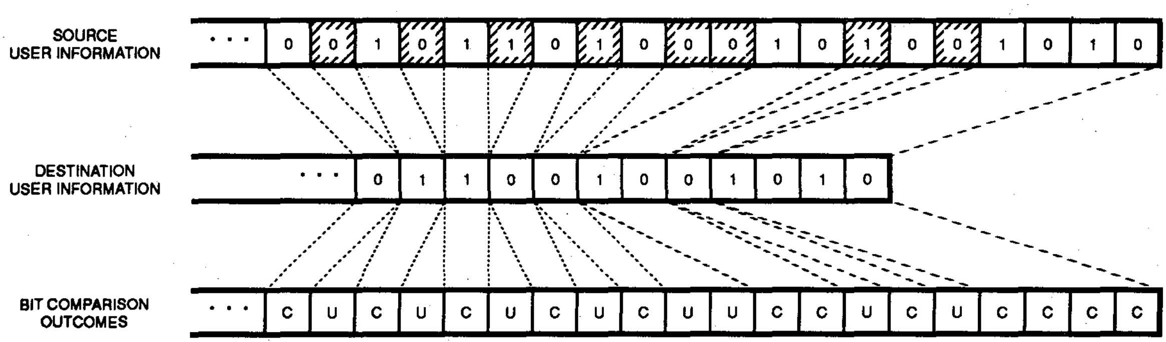
Figure 34. Scheme for finding appropriate user information window size.



a. Postulated (and Most Probable) BCO Sequence



b. BCO Sequence Identified by BITCOR when L(w)=3 or L(w)=2



c. BCO Sequence Identified by BITCOR when L(w)=1

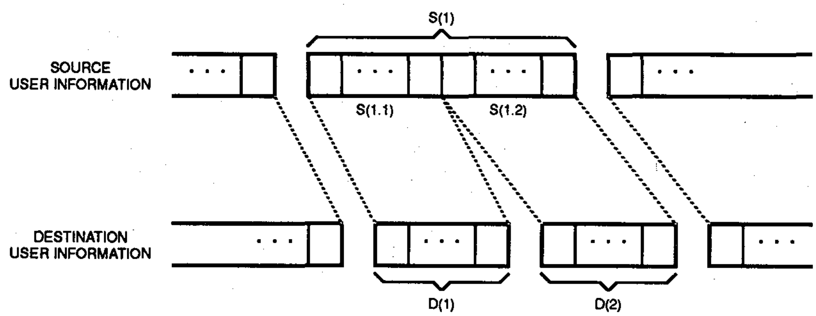
Figure 35. Example of poorly-isolated bit transfer failures.

Correlation procedures in **BITCOR** result in a correlation impasse (nonperformance) when L(w) > 3. These procedures identify the BCO sequence shown in Figure 35b when L(w) = 3 or L(w) = 2, and identify the BCO sequence shown in Figure 35c when L(w) = 1; neither of these sequences is the most probable BCO sequence $\hat{q}$. The performance of **BITCOR** correlation procedures exhibited in this example - incorrect performance for small values of L(w) and nonperformance for larger values of L(w) - is typical of cases in which bit transfer failures are poorly isolated or nonisolated.

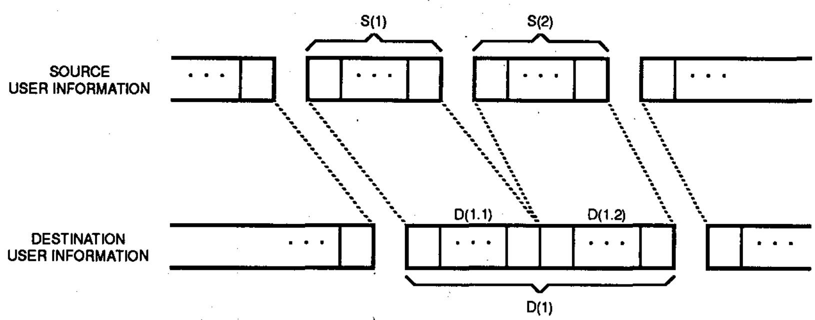
The distinction between well-isolated and poorly-isolated bit transfer failures is necessarily rather arbitrary. If a correlation impasse results when L(w) = 16, a possible strategy is to find and use the largest value of L(w) that does not result in an impasse. However, a BCO sequence obtained in this way should be accepted as the most probable sequence $\hat{q}$ only if further (operator) analysis of the data shows that alternative BCO sequences are clearly less probable.

4.2.1.2 Identification of Block Transfer Attempts

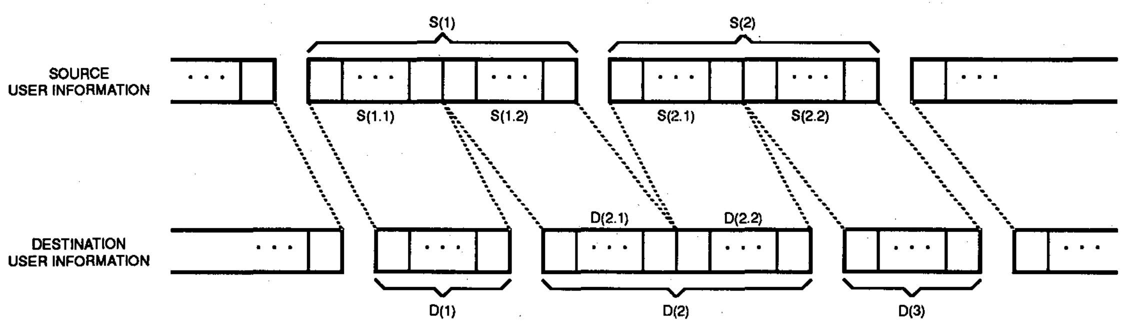
In some data communication sessions, a source block that contains no undelivered bits may fail to correspond to any destination block, or a destination block that contains no extra bits may fail to correspond to any source block. Figure 36 shows examples of such blocks observed during performance measurements conducted by ITS using the ARPANET (Wortendyke et al., 1982). In Figure 36a, user information input as a single source block S(1) is output as two consecutive destination blocks D(1) and D(2); destination blocks D(1) and D(2) correspond respectively to source bit strings S(1.1) and S(1.2). In this example, S(1) does not correspond to any destination block, and neither D(1) nor D(2) corresponds to any source block. In Figure 36b, user information input as two consecutive source blocks S(1) and S(2) is output as a single destination block D(1); source blocks S(1) and S(2) correspond respectively to destination bit strings D(1.1) and D(1.2). Neither S(1) nor S(2) corresponds to any destination block, and D(1) does not correspond to any source block. In Figure 36c, user information input as two consecutive source blocks S(1) and S(2) is output as three consecutive destination blocks D(1), D(2), and D(3); destination block D(1) corresponds to source bit string S(1.1), destination bit strings D(2.1) and D(2.2) respectively correspond to source bit strings S(1.2)



a. Fragmentation of Source Block



b. Consolidation of Source Blocks



c. Complex Reconfiguration of Source Blocks

Figure 36. Reconfiguration of user information block boundaries.

and S(2.1), and destination block D(3) corresponds to source bit string S(2.2). In this example, neither S(1) nor S(2) corresponds to any destination block, and neither D(1), D(2), nor D(3) corresponds to any source block. Noncorresponding blocks of the type illustrated in Figure 36a may also occur when data packets are partitioned at an interface between two packet-switched networks having different maximum data packet sizes.

An individual block transfer attempt is distinguished by the particular sequences of user information bits transmitted or received in the attempt. Examples in the preceding paragraph demonstrate the need to carefully formulate detailed specifications for associating a block transfer attempt with transmitted and received user information. Subroutine **BLKCOR** partitions the sequence of bit transfer attempts identified by **BITCOR** into a sequence of block transfer attempts so that a separate block transfer attempt is associated with

- each pair (S,D) where S is a source block containing one or more delivered bits and D is a string of destination bits consisting of (i) all bits that correspond to a bit in S and (ii) all strings of extra bits that satisfy specifications defined in the following paragraph,

- each source block consisting of undelivered bits, and

- each destination block consisting of extra bits and satisfying specifications defined in the following paragraph.

These three entities are the user information block analogues of pairs of corresponding source and destination bits, undelivered bits, and extra bits, respectively. A source block consisting of undelivered bits is called an <u>undelivered block</u>, and a destination block consisting of extra bits is called an <u>extra block</u>.

Source bits and nonextra destination bits are therefore associated with individual block transfer attempts as follows:

- All bits in a given source block are associated with the same block transfer attempt and no other source bits are associated with that attempt (i.e, all source bits associated with a given block transfer attempt are contained in a single source block).

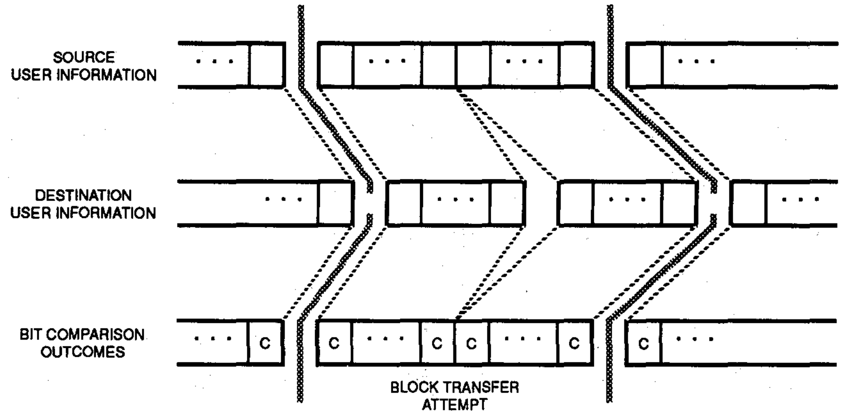
- A nonextra destination bit and the corresponding source bit are associated with the same block transfer attempt.

Extra destination bits are associated with individual block transfer attempts according to the following specifications:

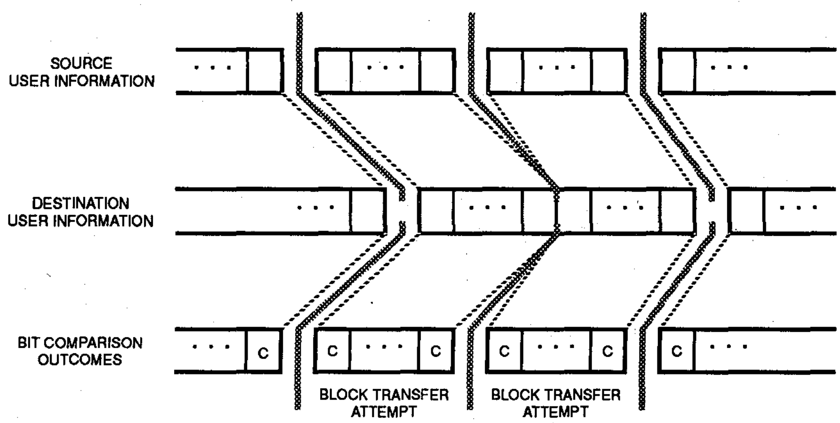
- If, within a destination block, a string of extra bits is preceded by one or more nonextra bits, then the extra bit string and the last nonextra bit preceding the string are associated with the same block transfer attempt. If, within a destination block, a string of extra bits at the beginning of the block is followed by one or more nonextra bits, then the extra bit string and the first nonextra bit following the string are associated with the same block transfer attempt.

- A destination block containing only extra bits is associated with a separate block transfer attempt if (i) the block precedes every nonextra bit in the batch, or (ii) the block follows every nonextra bit in the batch, or (iii) the last nonextra bit preceding the block and the first nonextra bit following the block are associated with different block transfer attempts. Otherwise, a destination block containing only extra bits is preceded by at least one nonextra bit and followed by at least one nonextra bit, where the last nonextra bit preceding the block and the first nonextra bit following the block are associated with a common block transfer attempt. In this case, the destination block, the last nonextra bit preceding the block, and the first nonextra bit following the block are associated with the same block transfer attempt.

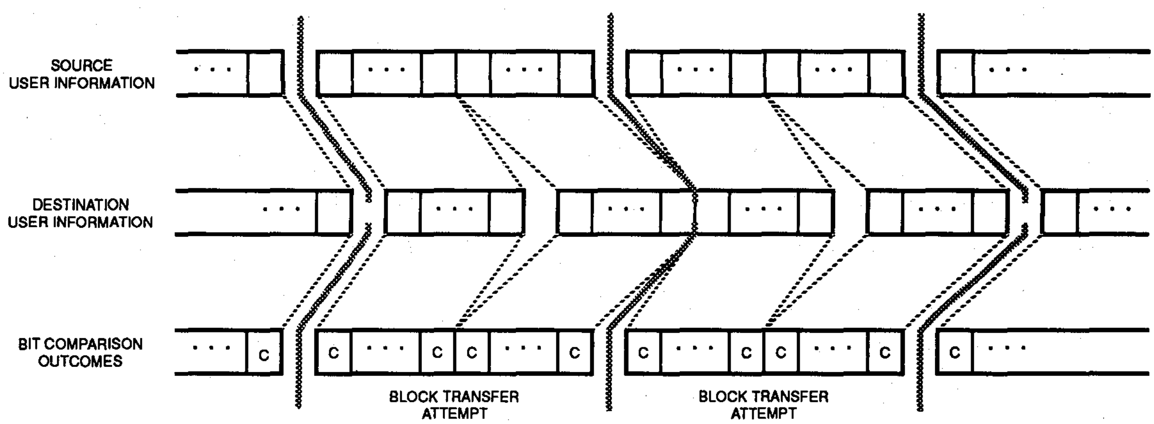
The preceding specifications for associating source and destination bits with an individual block transfer attempt are illustrated in Figures 37-40 In these figures, corresponding source and destination bit strings (and the associated BCOs) are indicated by thin dashed lines, and each nonextra destination bit is identical to the corresponding source bit. Figure 37 shows BCOs and block transfer attempts for the examples of noncorresponding source and destination blocks given in Figure 36. Each block transfer attempt in these examples is associated with a single source block and the corresponding destination bits. In Figure 37a, destination user information associated with a single block transfer attempt is contained in different destination blocks. In Figure 37b, destination user information associated with different block transfer attempts is contained in a single destination block. A combination of both cases is illustrated in Figure 37c.



a. Fragmentation of Source Block



b. Consolidation of Source Blocks



c. Complex Reconfiguration of Source Blocks

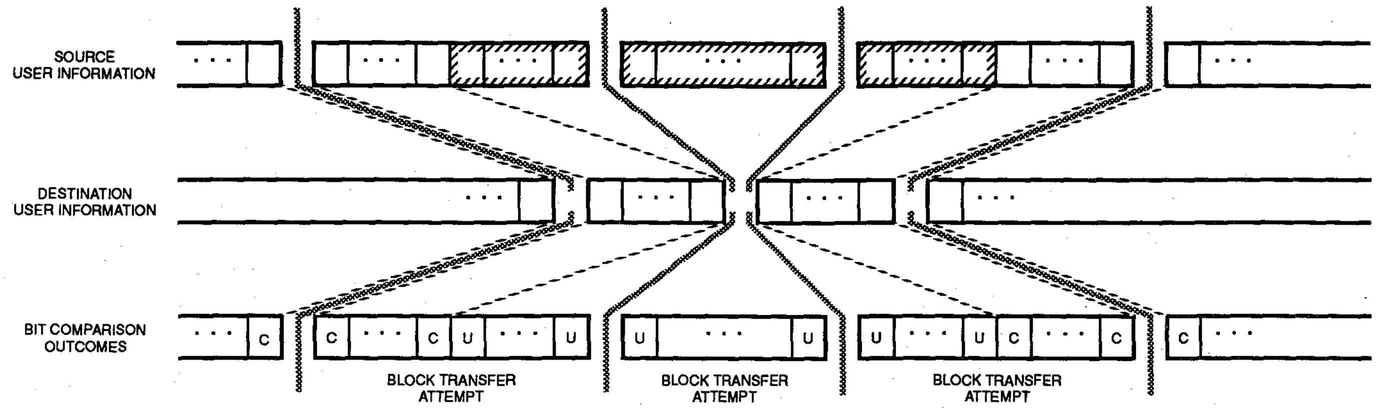
Figure 37. Block transfer attempts covering reconfigured user information block boundaries.

Figure 38 shows BCOs and block transfer attempts for two examples in which a string of undelivered bits (outlined by hachures) comprises the tail of one source block, all of the next source block, and the head of a third source block. Destination strings corresponding to the head of the first source block and the tail of the third source block are output as separate blocks in the first example and as a single block in the second example. In both examples, a separate block transfer attempt is associated with the source block consisting of undelivered bits. Each of the other indicated block transfer attempts is associated with a single source block and the corresponding destination bits.

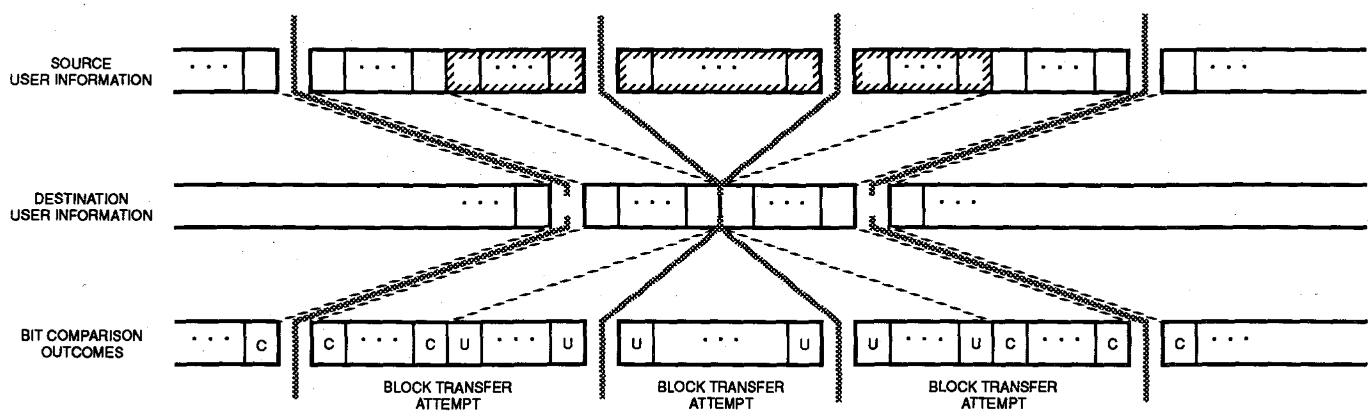
Figure 39 shows BCOs and block transfer attempts for three pairs of examples in which a destination block contains both extra and nonextra bits. A string of extra bits (outlined by hachures) occurs at the beginning of the block in the first pair of examples (Figure 39a), in the middle of the block in the second pair of examples (Figure 39b), and at the end of the block in the last pair of examples (Figure 39c). In the first example of each pair, the last nonextra bit preceding the extra string and the first nonextra bit following the extra string correspond to bits in the same source block. In the second example of each pair, these two nonextra bits correspond to bits in different source blocks.

In the first pair of examples, the extra bit string and the first nonextra bit following the string are associated with the same block transfer attempt. A single block transfer attempt is therefore associated with (i) the source block that contains the bit corresponding to the first nonextra bit following the string of extra bits, (ii) all destination bits that correspond to a bit in the indicated source block, and (iii) the string of extra bits. In the next two pairs of examples, the extra bit string and the last nonextra bit preceding the string are associated with the same block transfer attempt. A single block transfer attempt is therefore associated with (i) the source block that contains the bit corresponding to the last nonextra bit preceding the string of extra bits, (ii) all destination bits that correspond to a bit in the indicated source block, and (iii) the string of extra bits.

Figure 40 shows BCOs and block transfer attempts for some examples in which a destination block contains only extra bits. Such a block precedes all nonextra bits in Figure 40a and follows all nonextra bits in Figure 40b. In both

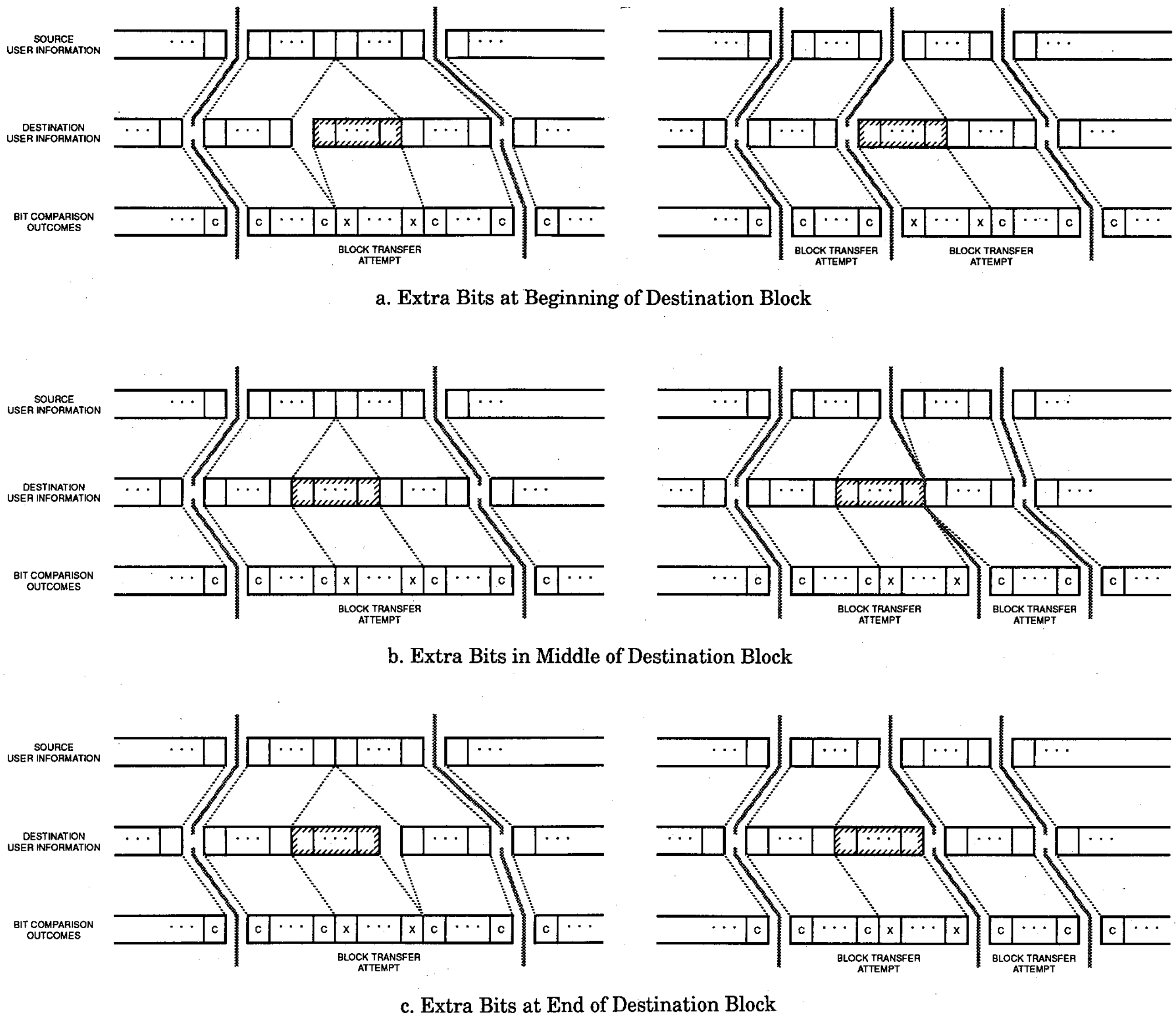


Example 1



Example 2

Figure 38. Block transfer attempts containing undelivered BCOs.

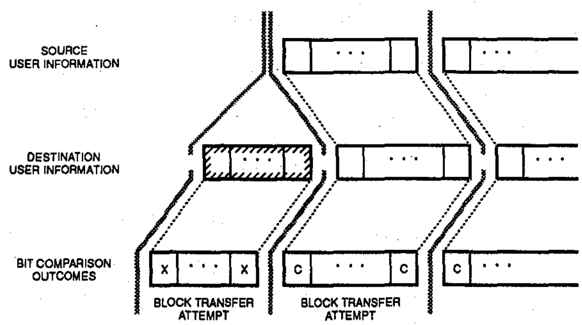


a. Extra Bits at Beginning of Destination Block

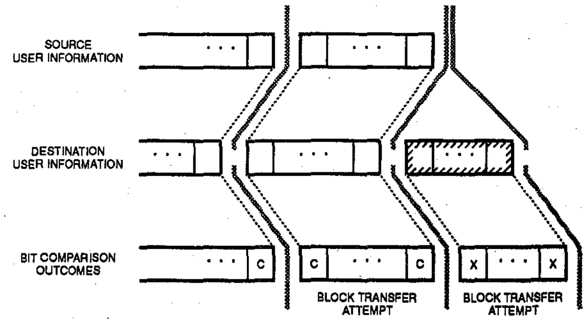
b. Extra Bits in Middle of Destination Block

c. Extra Bits at End of Destination Block

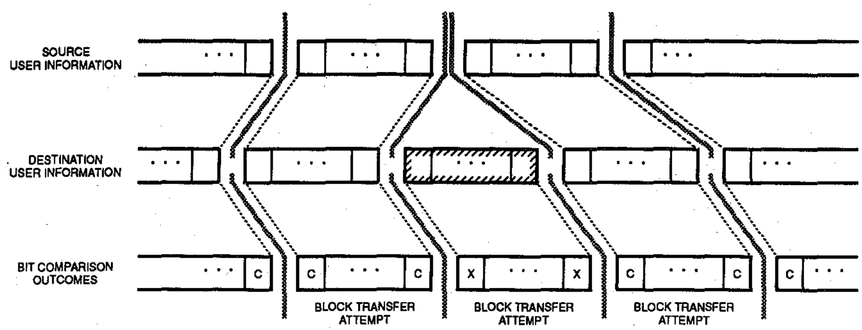
Figure. 39. Block transfer attempts containing both extra and nonextra BCOs.



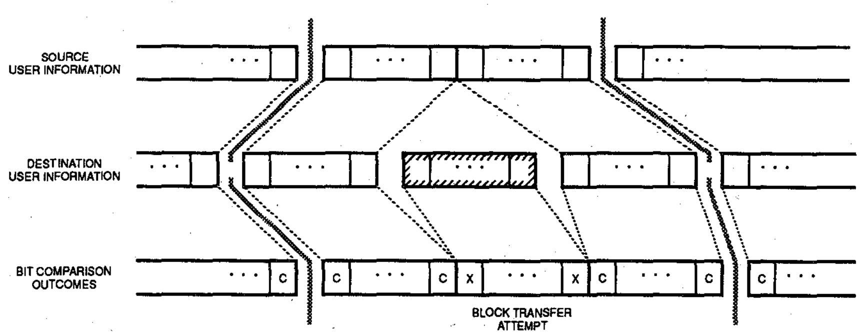
a. Initial Block of Extra Bits



b. Terminal Block of Extra Bits



c. Interior Block of Extra Bits Associated with a Separate Block Transfer Attempt



d. Interior Block of Extra Bits and Adjacent Nonextra Bits Associated with a Single Block Transfer Attempt

Figure 40. Block transfer attempts containing only extra BCOs.

examples, a separate block transfer attempt is associated with the block of extra bits. A destination block consisting of extra bits is both preceded and followed by nonextra bits in Figures 40c and 40d. In Figure 40c the last nonextra bit preceding the block of extra bits corresponds to a bit in one source block and the first nonextra bit following the block of extra bits corresponds to a bit in another source block. A separate block transfer attempt is therefore associated with the block of extra bits. In Figure 40d, the last nonextra bit preceding the block of extra bits and the first nonextra bit following the block of extra bits correspond to bits in the same source block. In this case, the block of extra bits and both adjacent nonextra bits are associated with the same block transfer attempt. A single block transfer attempt is therefore associated with (i) the source block that contains bits corresponding to the two nonextra bits adjacent to the block of extra bits, (ii) all destination bits that correspond to a bit in the indicated source block, and (iii) the block of extra bits.

a. BCO Assignment Algorithm

After **BITCOR** identifies a BCO, subroutine **BLKCOR** assigns it to the appropriate block transfer attempt in accordance with specifications presented earlier. The BCO assignment algorithm is outlined in Figure 41. A given (current) BCO is either included in the block transfer attempt that contains the preceding BCO or it is the first BCO in the next block transfer attempt. If the last correlated source bit is not the last bit in a source block, the current BCO is always included in the block transfer attempt that contains the preceding BCO. If the last correlated source bit is the last bit in a source block, or if there are no correlated source bits, the current BCO is assigned to a block transfer attempt as follows:

- A correct or incorrect BCO is included in the block transfer attempt that contains the preceding BCO if that attempt contains only extra BCOs and the last correlated destination bit is not the last bit in a destination block. Otherwise, a correct or incorrect BCO is the first BCO in the next block transfer attempt.
- An undelivered BCO is the first BCO in the next block transfer attempt.
- An extra BCO is included in the block transfer attempt that contains the preceding BCO if the last correlated destination

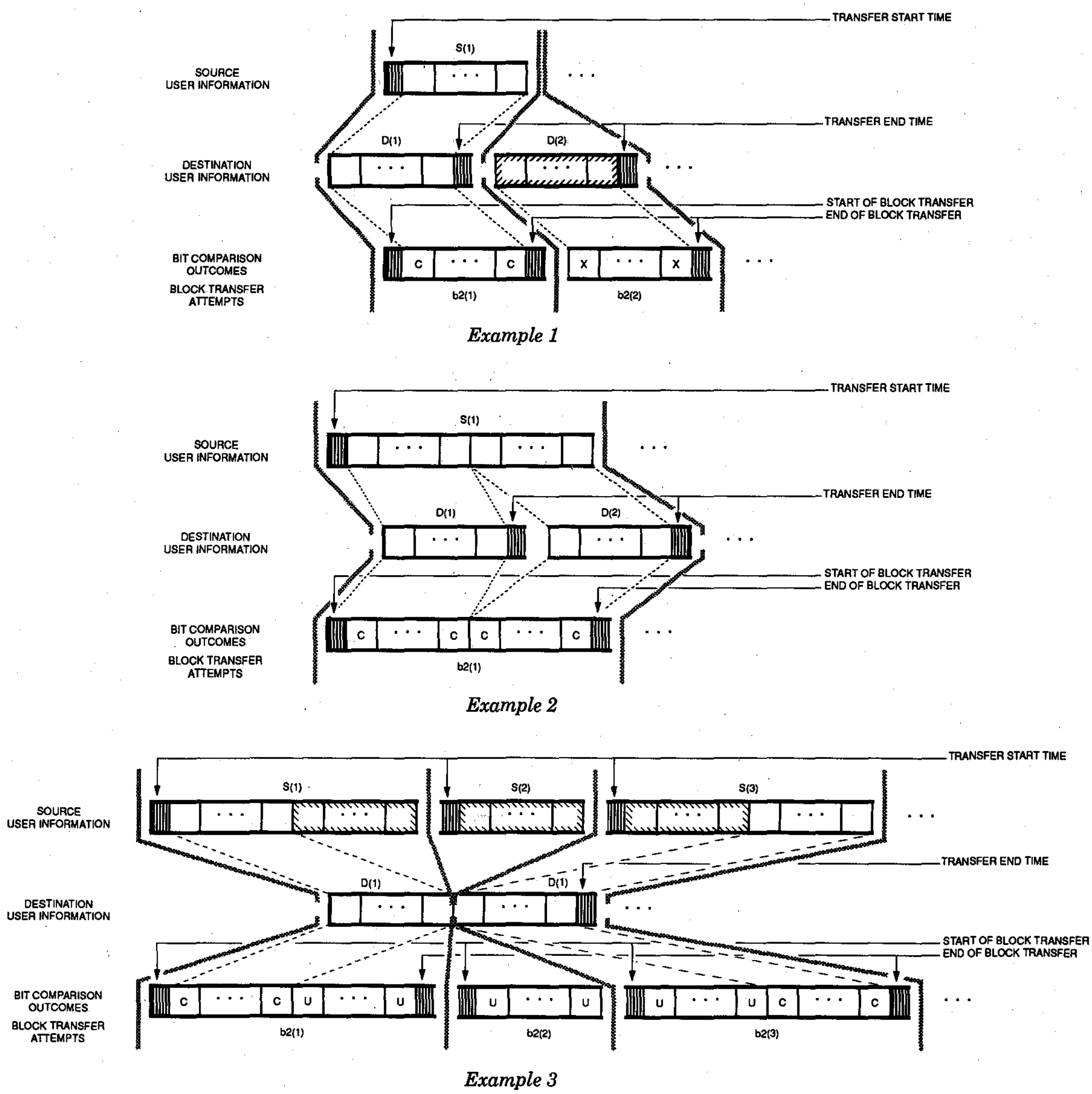


Figure 42. Examples of block transfer event time determination.

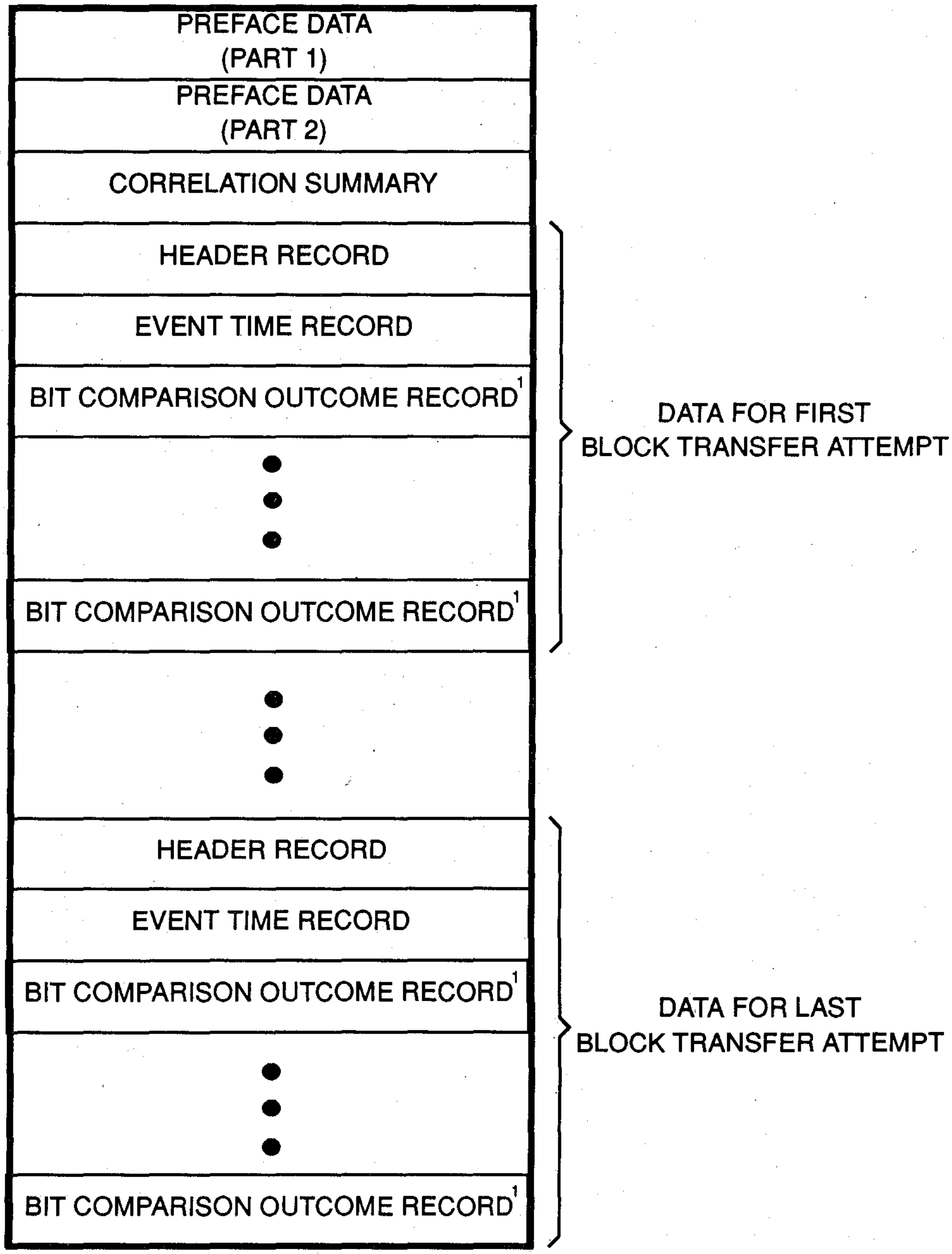
In Example 2, user information input as a single source block S(1) is output as two consecutive destination blocks D(1) and D(2); D(1) corresponds to the head of S(1), and D(2) corresponds to the tail of S(1). A single block transfer attempt is thus associated with S(1), D(1), and D(2). The End of Block Transfer event time for this attempt is the transfer end time for D(2). The transfer end time for D(1) is not used in performance assessment.

In Example 3, a string of undelivered bits (outlined by hachures) comprises the tail of source block S(1), all of source block S(2), and the head of source block S(3). The head of S(1) and the tail of S(3) are output as a single destination block D(1); the head of D(1) corresponds to the head of S(1), and the tail of D(1) corresponds to the tail of S(3). The first block transfer attempt is associated with S(1) and the head of D(1), the second attempt is associated with the undelivered block S(2), and the third attempt is associated with S(3) and the tail of D(1). The Start of Block Transfer event time for each attempt is the transfer start time for the associated source block. The End of Block Transfer event time for both the first attempt and the last attempt is the transfer end time for D(1); the End of Block Transfer event for the second attempt is not defined. Start of Block Transfer and End of Block Transfer event times for an attempt associated with an undelivered block are not relevant in user information transfer performance assessment.

4.2.1.3 Correlator Output File

Results of the correlation process are recorded in the correlator output file (**COR**). The record sequence in this file is shown in Figure 43, record formats are defined in Figure 44, and an example of a correlator output file is shown in Figure 45. Information in the correlation summary record is also listed in the assessment summary file (**SUM**) described in Section 5.

The correlator output file contains a <u>header record</u> and an <u>event time record</u> that summarize correlation results for each block transfer attempt identified by **BLKCOR**. The header record lists the numbers of BCOs in each category and the event time record lists event times for Start of Block Input, Start of Block Transfer, and End of Block Transfer. If a block transfer attempt includes one or more bit transfer failures (noncorrect BCOs), the event time record is followed by a sequence of <u>bit comparison outcome records</u> in which each successive BCO is represented by a <u>bit comparison outcome code</u>. Correct,



[1] Record is included only when block transfer attempt contains one or more bit comparison outcomes that represent bit transfer failures.

Figure 43. Record sequence in correlator output file.

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
PREFACE DATA (PART 1):		
1-32	A32	FILE DESCRIPTOR
PREFACE DATA (PART 2):		
1-64	A64	BATCH IDENTIFIER
CORRELATION SUMMARY:		
1-8	F8.0	NUMBER OF BIT COMPARISON OUTCOMES IN FILE
9-16	F8.0	NUMBER OF BLOCK TRANSFER ATTEMPTS IN FILE
17-24	F8.0	NUMBER OF 'CORRECT' BIT COMPARISON OUTCOMES IN FILE
25-32	F8.0	NUMBER OF 'INCORRECT' BIT COMPARISON OUTCOMES IN FILE
33-40	F8.0	NUMBER OF 'UNDELIVERED' BIT COMPARISON OUTCOMES IN FILE
41-48	F8.0	NUMBER OF 'EXTRA' BIT COMPARISON OUTCOMES IN FILE
49-56	F8.0	NUMBER OF SOURCE USER INFORMATION BITS CORRELATED
57-64	F8.0	NUMBER OF SOURCE USER INFORMATION BLOCKS CORRELATED
65-72	F8.0	NUMBER OF DESTINATION USER INFORMATION BITS CORRELATED
73-80	F8.0	NUMBER OF DESTINATION USER INFORMATION BLOCKS CORRELATED

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
HEADER RECORD:		
1-8	F8.0	INDEX OF BLOCK TRANSFER ATTEMPT
9-16	F8.0	NUMBER OF BIT COMPARISON OUTCOMES IN BLOCK TRANSFER ATTEMPT
17-24	F8.0	NUMBER OF 'CORRECT' BIT COMPARISON OUTCOMES IN BLOCK TRANSFER ATTEMPT
25-32	F8.0	NUMBER OF 'INCORRECT' BIT COMPARISON OUTCOMES IN BLOCK TRANSFER ATTEMPT
33-40	F8.0	NUMBER OF 'UNDELIVERED' BIT COMPARISON OUTCOMES IN BLOCK TRANSFER ATTEMPT
41-48	F8.0	NUMBER OF 'EXTRA' BIT COMPARISON OUTCOMES IN BLOCK TRANSFER ATTEMPT
EVENT TIME RECORD:		
1-16	D16.9	BLOCK INPUT START TIME (SECONDS AFTER REFERENCE TIME)
17-32	D16.9	BLOCK TRANSFER START TIME (SECONDS AFTER REFERENCE TIME)
33-48	D16.9	BLOCK TRANSFER END TIME (SECONDS AFTER REFERENCE TIME
BIT COMPARISON OUTCOME RECORD:		
1	I1	BIT COMPARISON OUTCOME CODE
2	I1	BIT COMPARISON OUTCOME CODE
		⋮
80	I1	BIT COMPARISON OUTCOME CODE

Figure 44. Record formats in correlator output file.

```
CORRELATOR OUTPUT
NTIA-ITS (Boulder)                                            1424
 81944.     80.  81888.     8.     24.     24.  81920.     80.  81920.     80.
     1.   1024.   1024.     0.      0.      0.
 0.366947690D+05 0.366947690D+05 0.366955540D+05
     2.   1024.   1024.     0.      0.      0.
 0.366948970D+05 0.366948970D+05 0.366957340D+05
     3.   1024.   1024.     0.      0.      0.
 0.366950350D+05 0.366950350D+05 0.366967030D+05
     4.   1024.   1024.     0.      0.      0.
 0.366951680D+05 0.366951680D+05 0.366968870D+05
     5.   1024.   1024.     0.      0.      0.
 0.366953050D+05 0.366953050D+05 0.366970470D+05
     6.   1024.   1024.     0.      0.      0.
 0.366954380D+05 0.366954380D+05 0.366972290D+05
     7.   1024.   1024.     0.      0.      0.
 0.366955750D+05 0.366955750D+05 0.366974350D+05
     8.   1024.   1024.     0.      0.      0.
 0.366957080D+05 0.366957080D+05 0.366975940D+05
     9.   1024.   1024.     0.      0.      0.
 0.366958450D+05 0.366958450D+05 0.366977780D+05
    10.   1024.   1016.     8.      0.      0.
 0.366959780D+05 0.366959780D+05 0.366979620D+05
11111111111111111111111111111111111111111111111111111111111111111111111112122122
11121112112111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111110000000000000000
    11.   1024.   1024.     0.      0.      0.
 0.366961120D+05 0.366961120D+05 0.366981230D+05
    12.   1024.   1024.     0.      0.      0.
 0.366962490D+05 0.366962490D+05 0.366982810D+05
    13.   1024.   1024.     0.      0.      0.
 0.366963820D+05 0.366963820D+05 0.366984650D+05
    14.   1024.   1024.     0.      0.      0.
 0.366965200D+05 0.366965200D+05 0.366986490D+05
    15.   1024.   1024.     0.      0.      0.
 0.366966520D+05 0.366966520D+05 0.366988070D+05
    16.   1024.   1024.     0.      0.      0.
 0.366967900D+05 0.366967900D+05 0.366989920D+05
    17.   1024.   1024.     0.      0.      0.
 0.366969220D+05 0.366969220D+05 0.366991730D+05
    18.   1024.   1024.     0.      0.      0.
 0.366978210D+05 0.366978210D+05 0.366993790D+05
    19.   1048.   1000.     0.     24.     24.
 0.366979650D+05 0.366979650D+05 0.366995870D+05
11111111111111111111111111133333333333333333333333311111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111111111111111111111111111111111111111111111111111111111111111111
11111111111111111144444444444444444444444411111111111111111111111111111111111111
11111110000000000000000000000000000000000000000000000000000000000000000000000000
    20.   1024.   1024.     0.      0.      0.
 0.366980970D+05 0.366980970D+05 0.367004820D+05
    21.   1024.   1024.     0.      0.      0.
 0.366982350D+05 0.366982350D+05 0.367006660D+05
                  ...
                  ...
                  ...
    79.   1024.   1024.     0.      0.      0.
 0.367124060D+05 0.367124060D+05 0.367146620D+05
    80.   1024.   1024.     0.      0.      0.
 0.367125380D+05 0.367125380D+05 0.367148510D+05
```

Figure 45. Example of a correlator output file.

incorrect, undelivered, and extra BCOs are indicated by the codes 1, 2, 3, and 4, respectively. Any unused space at the end of the last BCO record for a block transfer attempt contains ASCII-zero fill.

4.2.2 Determination of Bit and Block Transfer Outcomes

Subroutine **TRANSF** determines outcomes of bit and block transfer attempts in accordance with the definitions given in ANS X3.102. Block transfer outcomes may be characterized as follows:

> Successful Block Transfer occurs if (i) Start of Block Transfer occurs, (ii) End of Block Transfer occurs on or before the performance deadline, and (iii) received user information is identical to transmitted user information.
>
> Incorrect Block occurs if (i) Start of Block Transfer occurs, (ii) End of Block Transfer occurs on or before the performance deadline, and (iii) received user information is not identical to transmitted user information.
>
> Lost Block occurs if (i) Start of Block Transfer occurs, (ii) End of Block Transfer does not occur on or before the performance deadline, and (iii) responsibility for the failure is attributed to system nonperformance.
>
> Refused Block occurs if (i) Start of Block Transfer occurs, (ii) End of Block Transfer does not occur on or before the performance deadline, and (iii) responsibility for the failure is attributed to user nonperformance.
>
> Extra Block occurs if (i) Start of Block Transfer does not occur and (ii) End of Block Transfer occurs.

A block transfer attempt whose outcome is Successful Block Transfer contains only correct BCOs. All bits transmitted by the source user are delivered without error to the intended destination user no later than the block transfer performance deadline.

In a block transfer attempt whose outcome is Incorrect Block, at least some of the bits transmitted by the source user are delivered to the intended destination user; all bits received are delivered no later than the performance deadline. Consequently, the block transfer attempt contains at least one correct or incorrect BCO. Because the sequence of received bits is not identical to the sequence of transmitted bits, the block transfer attempt contains at least one noncorrect BCO. The noncorrect BCOs may consist of any combination of incorrect, undelivered, or extra BCOs.

The Lost Block outcome may occur in two distinct ways: (i) none of the source bits transmitted in the block transfer attempt is delivered to the intended destination user and responsibility for the loss is attributed to system nonperformance or (ii) at least some of the transmitted source bits are delivered to the intended destination user subsequent to the block transfer performance deadline and responsibility for the excessive delay is attributed to system nonperformance. In case (i), the block transfer attempt contains only undelivered BCOs and End of Block Transfer does not occur. In case (ii), the block transfer attempt contains at least one correct or incorrect BCO and End of Block Transfer occurs. In the latter case, the block transfer attempt may also contain undelivered or extra BCOs.

When system nonperformance is replaced by user nonperformance in the preceding paragraph, the discussion applies to Refused Block outcomes. In accordance with ANS X3.102, a block transfer attempt whose outcome is Refused Block is excluded from the set of trials used to estimate values of block transfer performance parameters.

A block transfer attempt whose outcome is Extra Block contains only extra BCOs (no bits are transmitted by the source user). Because Start of Block Transfer does not occur, the block transfer performance deadline cannot be evaluated. Though End of Block Transfer occurs, the event time is not relevant for performance assessment.

Bit transfer outcomes may be characterized as follows:

<u>Successful Bit Transfer</u> occurs if (i) a source bit is transmitted, (ii) a corresponding destination bit is received, (iii) the transmitted and received bits have the same binary value, and (iv) the bit transfer attempt is contained in a block transfer attempt in which End of Block Transfer occurs no later than the performance deadline.

<u>Incorrect Bit</u> occurs if (i) a source bit is transmitted, (ii) a corresponding destination bit is received, (iii) the transmitted and received bits have different binary values, and (iv) the bit transfer attempt is contained in a block transfer attempt in which End of Block Transfer occurs no later than the performance deadline.

<u>Lost Bit</u> occurs if (i) a source bit is transmitted, (ii) a corresponding destination bit is not received or the bit transfer

attempt is contained in a block transfer attempt in which End of Block Transfer does not occur on or before the performance deadline, and (iii) responsibility for the failure is attributed to system nonperformance.

Refused Bit occurs if the bit transfer attempt is contained in a block transfer attempt whose outcome is Refused Block.

Extra Bit occurs if (i) a destination bit is received, (ii) a corresponding source bit is not transmitted, and (iii) the bit transfer attempt is not contained in a block transfer attempt whose outcome is Refused Block.

A bit transfer attempt whose outcome is Successful Bit Transfer is a correct BCO in a block transfer attempt whose outcome is Successful Block Transfer or Incorrect Block. A bit transfer attempt whose outcome is Incorrect Bit is an incorrect BCO in a block transfer attempt whose outcome is Incorrect Block. A bit transfer attempt whose outcome is Lost Bit is (i) an undelivered BCO in a block transfer attempt whose outcome is Incorrect Block or (ii) a correct, incorrect, or undelivered BCO in a block transfer attempt whose outcome is Lost Block. A bit transfer attempt whose outcome is Extra Bit is an extra BCO in a block transfer attempt whose outcome is Incorrect Block, Lost Block, or Extra Block. In accordance with ANS X3.102, a bit transfer attempt whose outcome is Refused Bit is excluded from the set of trials used to estimate values of bit transfer performance parameters.

4.2.2.1 Outcome Determination Scheme

The scheme used in subroutine **TRANSF** to determine outcomes of bit and block transfer attempts is outlined by the flowchart in Figure 46. Block transfer outcomes are determined by three procedures corresponding respectively to the three columns of decision symbols in the figure.

The first procedure

- identifies block transfer attempts in which Start of Block Transfer does not occur and assigns these to the Extra Block outcome category (the block transfer attempt contains only extra BCOs),
- identifies block transfer attempts in which End of Block Transfer does not occur (the block transfer attempt contains only undelivered BCOs),

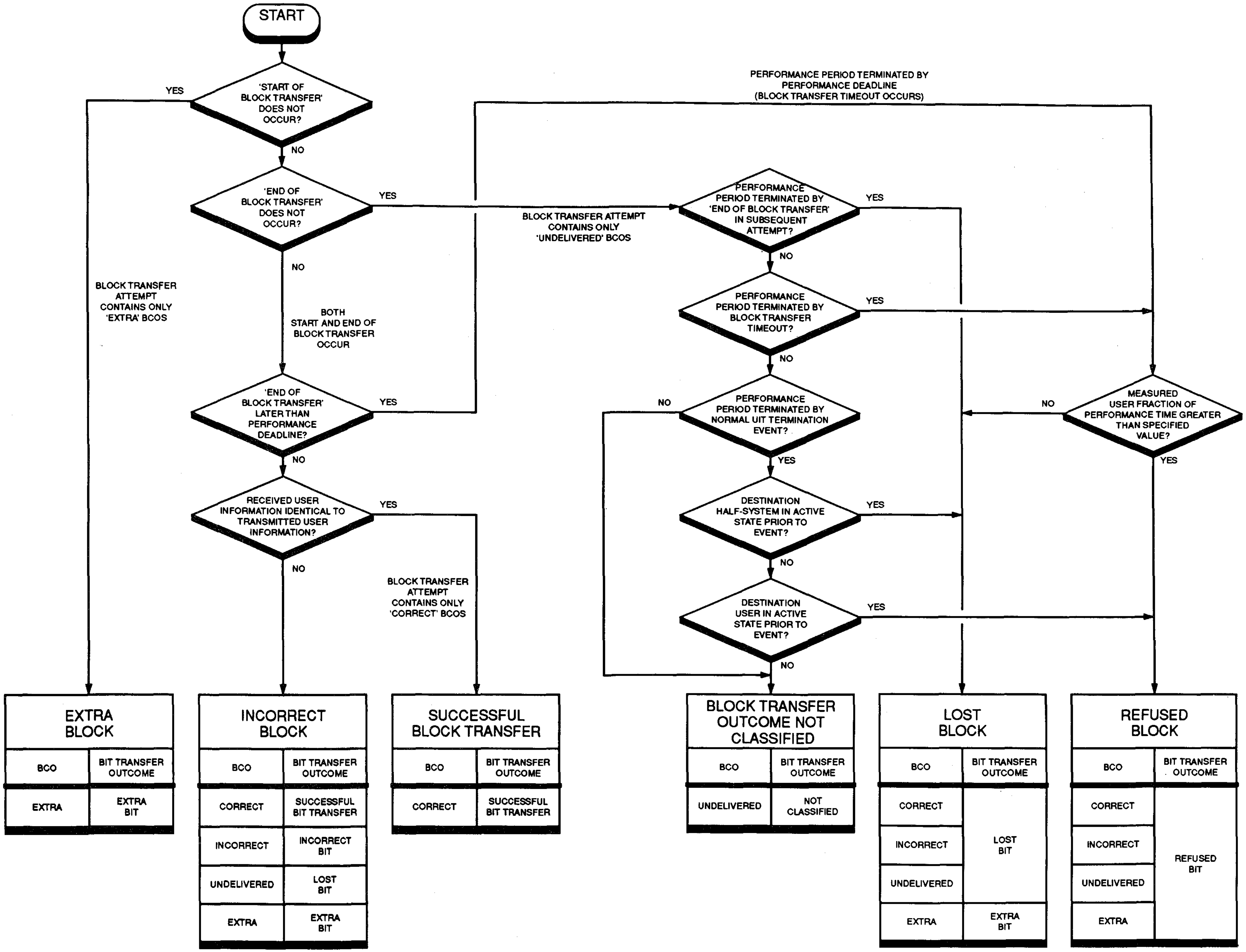


Figure 46. Scheme for determining block transfer and bit transfer outcomes.

- identifies block transfer attempts in which End of Block Transfer is later than the performance deadline (block transfer timeout occurs), and

- identifies block transfer attempts in which End of Block Transfer is earlier than or coincident with the performance deadline and assigns these to (i) the Successful Block Transfer outcome category if received user information is identical to transmitted user information or (ii) the Incorrect Block outcome category if received user information is not identical to transmitted user information.

The second outcome determination procedure analyzes block transfer attempts in which End of Block Transfer does not occur. In this procedure, the end of the performance period for such an attempt corresponds to whichever occurs first after Start of Block Transfer:

- End of Block Transfer in a subsequent attempt,

- the performance deadline (for the current block transfer attempt), or

- a user information transfer (UIT) termination event at the destination interface.

If the performance period for a block transfer attempt that contains only undelivered BCOs is terminated by End of Block Transfer in a subsequent attempt, responsibility for the block transfer failure is attributed to system nonperformance and the block transfer outcome is Lost Block.

While user information transfer (UIT) is in progress at an interface, both local entities are in a committed state. If, in an event at an interface where user information transfer is in progress, an entity undergoes a transition from a committed state, the event is regarded by **TRANSF** as the end of user information transfer activities at the interface. Such an event is called a UIT termination event; a UIT termination event is normal if both local entities undergo transitions from a committed state to a closing state and is anomalous otherwise.

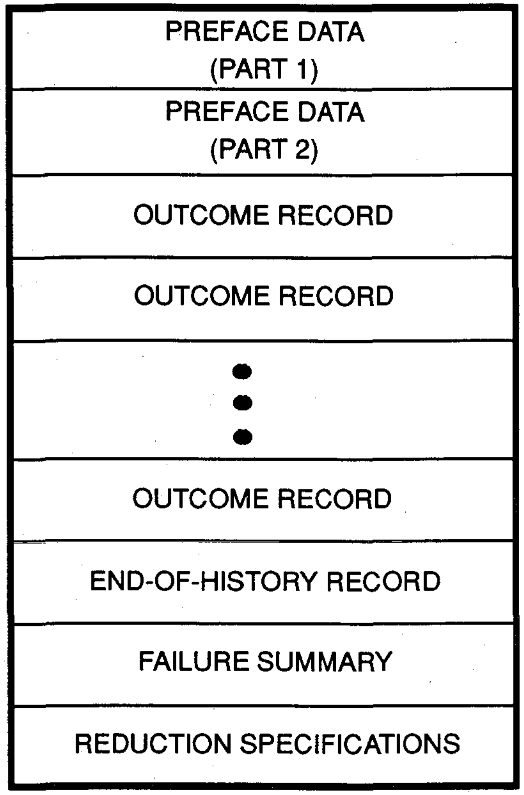
A normal UIT termination event is generated by whichever local entity is in the active ancillary state prior to the event. If the performance period for a block transfer attempt that contains only undelivered BCOs is ended by a normal UIT termination event generated by the destination half-system, responsibility for the block transfer failure is attributed to system nonperformance and the block transfer outcome is Lost Block. If the period is ended by a normal UIT termination event generated by the destination user, responsibility for the block transfer failure is attributed to user nonperformance and the block transfer outcome is Refused Block. If the performance period is ended by any other UIT termination event, the event history is anomalous (or erroneous) and the outcome of the block transfer attempt is not classified.

The final outcome determination procedure outlined in Figure 46 treats block transfer attempts in which the performance period is terminated by block transfer timeout. The procedure evaluates the user fraction of performance time for the period. If the measured fraction exceeds the specified value for User Fraction of Block Transfer Time, responsibility for the excessive delay is attributed to user nonperformance and the outcome of the block transfer attempt is Refused Block. Otherwise, that responsibility is attributed to system nonperformance and the block transfer outcome is Lost Block.

When the outcome of a block transfer attempt has been determined, outcomes of the associated bit transfer attempts are determined by a procedure that assigns the corresponding BCOs to bit transfer outcome categories according to the scheme shown in Figure 46.

4.2.2.2 Bit and Block Transfer Outcome Files

Outcomes of individual block transfer attempts are recorded in the block transfer outcome file (B2O) as they are determined. The record sequence in this file is shown in Figure 47a, record formats are defined in Figure 47b, and an example of a block transfer outcome file is shown in Figure 48. Each block transfer attempt identified by **TRANSF** is represented by an <u>outcome record</u> in the block transfer outcome file. The outcome record for a successful block transfer attempt contains overall and user performance times, whereas the record for an unsuccessful attempt contains a negative <u>block transfer outcome code</u>. Incorrect Block, Lost Block, Extra Block, and Refused Block outcomes are indicated by -1, -3, -4, and -5, respectively. Block transfer attempts whose outcomes are not classified are indicated by -9. The final outcome record in the file is followed by an <u>end-of-history record</u>. The latter contains an <u>end-of-history code</u> (-30) and has the same format as the outcome record for an unsuccessful block transfer attempt. The subsequent <u>failure summary record</u> lists the number of trials in the measurement sample and lists, for each system-responsible failure outcome category, the observed numbers of failures and pairs of consecutive failures.



a. Record Sequence

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
PREFACE DATA (PART 1):		
1-32	A32	FILE DESCRIPTOR
PREFACE DATA (PART 2):		
1-64	A64	BATCH IDENTIFIER
OUTCOME RECORD (SUCCESSFUL PERFORMANCE):		
1-8	F8.3	OVERALL PERFORMANCE TIME FOR BLOCK TRANSFER ATTEMPT (SECONDS)
9-16	F8.3	USER PERFORMANCE TIME FOR BLOCK TRANSFER ATTEMPT (SECONDS)
OUTCOME RECORD (UNSUCCESSFUL PERFORMANCE):		
1-8	F8.0	OUTCOME CODE FOR UNSUCCESSFUL BLOCK TRANSFER ATTEMPT
9-16	F8.0	OUTCOME CODE FOR UNSUCCESSFUL BLOCK TRANSFER ATTEMPT
END-OF-HISTORY RECORD:		
1-8	F8.0	END-OF-HISTORY CODE (-30)
9-16	F8.0	END-OF-HISTORY CODE (-30)
FAILURE SUMMARY:		
1-8	F8.0	NUMBER OF BLOCK TRANSFER ATTEMPTS IN MEASUREMENT SAMPLE
9-16	F8.0	NUMBER OF 'INCORRECT BLOCK' OUTCOMES
17-24	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'INCORRECT BLOCK' OUTCOMES
25-32	F8.0	NUMBER OF 'LOST BLOCK' OUTCOMES
33-40	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'LOST BLOCK' OUTCOMES
41-48	F8.0	NUMBER OF 'EXTRA BLOCK' OUTCOMES
49-56	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'EXTRA BLOCK' OUTCOMES
REDUCTION SPECIFICATIONS:		
1-16	E16.8	SPECIFIED VALUE OF BLOCK TRANSFER TIME (SECONDS)
17-32	E16.8	SPECIFIED VALUE OF USER FRACTION OF BLOCK TRANSFER TIME

b. Record Formats

Figure 47. Record sequence and record formats in block transfer outcome file.

```
BLOCK TRANSFER OUTCOME
NTIA-ITS (Boulder)                                                1424
   0.785   0.000
   0.837   0.026
   1.668   0.052
   1.719   0.078
   1.742   0.104
   1.791   0.130
   1.860   0.135
   1.886   0.156
   1.933   0.156
     -1.     -1.
   2.011   0.208
   2.032   0.234
   2.083   0.260
   2.129   0.286
   2.155   0.312
   2.202   0.312
   2.251   0.312
   1.558   0.208
     -1.     -1.
   2.385   0.234
   2.431   0.234
   2.457   0.234
   2.503   0.234
   2.531   0.260
        ...
        ...
        ...
   1.793   0.245
   1.843   0.260
   2.210   0.260
   2.256   0.260
   2.313   0.260
    -30.    -30.
     80.      2.       0.       0.       0.       0.       0.
  0.30000000E+01  0.50000000E+00
```

Figure 48. Example of a block transfer outcome file.

The final reduction specifications record in the block transfer outcome file contains specified values of Block Transfer Time and User Fraction of Block Transfer Time used in outcome determination. Information in the block transfer outcome file enables the statistical analysis program **STAR** (described in Volume 5 of this report) to calculate estimated values and their confidence limits for all block transfer performance parameters defined in ANS X3.102 (with the exception of Misdelivered Block Probability).

The bit transfer outcome file (**B1O**) contains a summary of bit transfer failures observed by subroutine **TRANSF**. The record sequence in this file is shown in Figure 49a, record formats are defined in Figure 49b, and an example of a bit transfer outcome file is illustrated in Figure 49c. The failure summary record lists the number of bit transfer trials in the measurement sample and lists, for each system-responsible failure outcome category, the observed numbers of failures and pairs of consecutive failures. Information in the bit transfer outcome file enables **STAR** to calculate estimated values and their confidence limits for all bit transfer failure probabilities defined in ANS X3.102 (with the exception of Misdelivered Bit Probability).

Outcomes of individual bit transfer attempts are not recorded in the bit transfer outcome file. However, the outcome of any particular bit transfer attempt can be obtained as follows:

1. Locate the corresponding BCO in the correlator output file.
2. Obtain the outcome of the associated block transfer attempt from the block transfer outcome file.
3. Use the scheme in Figure 46 to assign the BCO to the proper bit transfer outcome category.

4.2.2.3 Block Transfer Outcomes and Mislocation of Undelivered and Extra BCOs

As described in parts b.2 and b.3 of Section 4.2.1, whenever one or more bits at the beginning of a well-isolated string S(U) of undelivered bits or a well-isolated string D(X) of extra bits are identical to bits that immediately follow S(U) or D(X), then **BITCOR** correlation procedures mislocate the associated string of noncorrect BCOs within the overall BCO sequence. If S(U) ends at or sufficiently near the end of a source block, or if D(X) ends at or sufficiently near the end of a destination block, mislocation of the associated BCOs may result in erroneous block transfer outcomes.

PREFACE DATA (PART 1)
PREFACE DATA (PART 2)
FAILURE SUMMARY

a. Record Sequence

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
PREFACE DATA (PART 1):		
1-32	A32	FILE DESCRIPTOR
PREFACE DATA (PART 2):		
1-64	A64	BATCH IDENTIFIER
FAILURE SUMMARY:		
1-8	F8.0	NUMBER OF BIT TRANSFER ATTEMPTS IN MEASUREMENT SAMPLE
9-16	F8.0	NUMBER OF 'INCORRECT BIT' OUTCOMES
17-24	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'INCORRECT BIT' OUTCOMES
25-32	F8.0	NUMBER OF 'LOST BIT' OUTCOMES
33-40	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'LOST BIT' OUTCOMES
41-48	F8.0	NUMBER OF 'EXTRA BIT' OUTCOMES
49-56	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'EXTRA BIT' OUTCOMES

b. Record Formats

```
BIT TRANSFER OUTCOME
NTIA-ITS (Boulder)                                              1424
  81944.      8.      2.     24.     23.     24.     23.
```

c. Example File

Figure 49. Bit transfer outcome file.

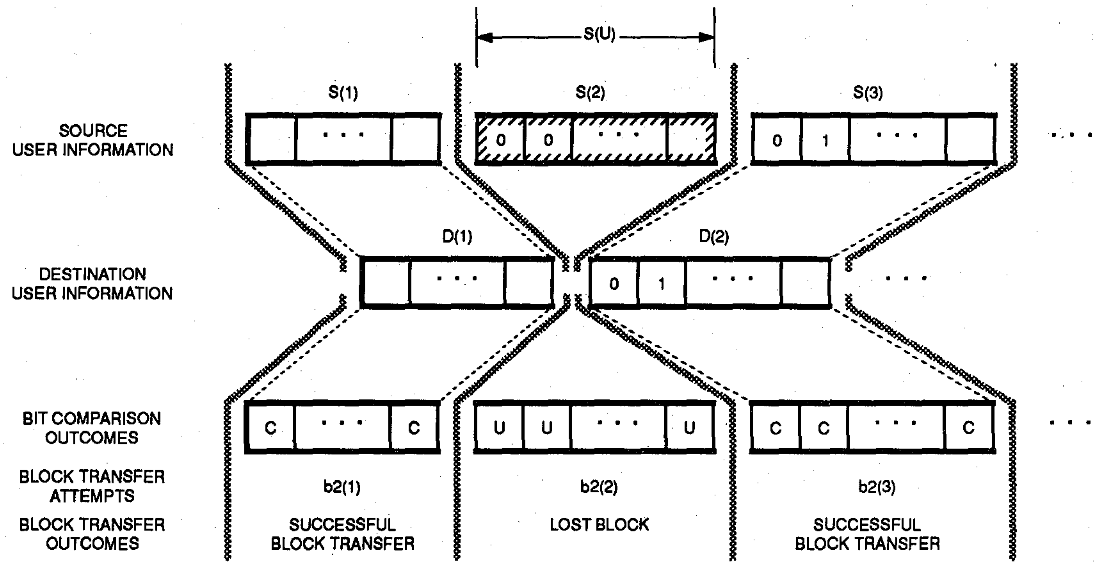
The problem is illustrated in Figure 50. In Figure 50a, a source block S(2) consists of a string S(U) of undelivered bits and is respectively preceded and followed by correctly delivered source blocks S(1) and S(3); the first bit in S(U) is identical to the first bit that immediately follows S(U) - i.e., the initial bit in S(3). Associated BCOs and block transfer attempts are indicated in the figure. However, **BITCOR** correlation procedures identify the undelivered string S'(U) and the associated BCOs shown in Figure 50b, and **BLKCOR** identifies the indicated block transfer attempts. The outcomes of the second and third block transfer attempts in the postulated case are Lost Block and Successful Block Transfer, respectively, whereas the outcomes of the second and third block transfer attempts identified by the **BITCOR** correlation procedures are both Incorrect Block.

When a small number of BCOs at the end of a string of undelivered or extra BCOs are assigned by **BITCOR** to a succeeding block transfer attempt, user information should be examined (subsequent to the reduction run) to identify cases where this spillover can be attributed to mislocation of the noncorrect BCO string. Erroneous block transfer outcomes determined by **TRANSF** in such cases may then be corrected.

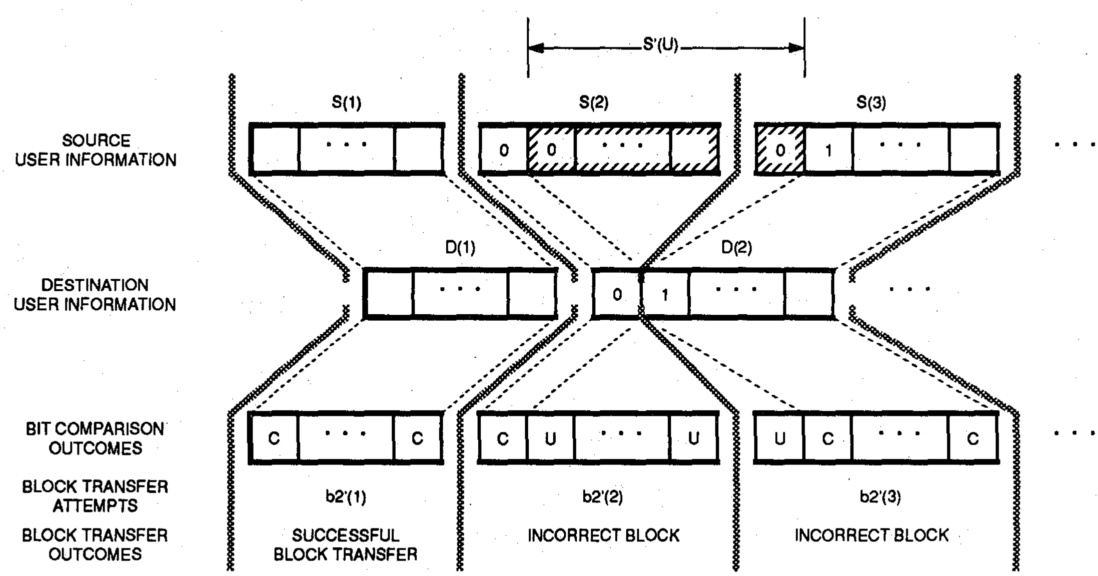
For some data communication systems, a configuration of undelivered or extra BCOs such as that shown in Figure 50b is improbable or impossible owing to characteristics of the user information transfer process. For instance, the number of undelivered bits in a source block or the number of extra bits in a destination block may be restricted to integer multiples of some information unit (e.g., an ASCII character). For these systems, the reexamination of data indicated in the preceding paragraph is essential.

4.2.3 Availability and Throughput Transfer Samples

Subroutine **TTSAMP**, a procedure invoked by **TRANSF**, selects transfer samples, called <u>availability transfer samples</u>, for the measurement of Transfer Denial Probability and determines their outcomes in accordance with ANS X3.102 and ANS X3.141. **TTSAMP** also selects a transfer sample, called a <u>throughput transfer sample</u>, for the measurement of long-term throughput parameters (User Information Bit Transfer Rate and User Fraction of Input/Output Time) and evaluates a set of performance descriptors for the sample. Each call to **TTSAMP** carries out transfer sample processing for a specified block transfer attempt.



a. Postulated Configuration of Undelivered Bits



b. Configuration of Undelivered Bits Identified by BITCOR

Figure 50. Effect of mislocated undelivered BCOs on block transfer outcomes.

This section describes (i) some general transfer sample concepts, (ii) criteria used by **TTSAMP** in selecting availability and throughput transfer samples, (iii) determination of availability transfer sample outcomes, and (iv) output files containing results of availability and throughput transfer sample processing.

4.2.3.1 Transfer Sample Concepts

Several basic concepts that underlie transfer sample processing performed by **TTSAMP** are illustrated in Figure 51, which depicts a sequence of block transfer attempts (indicated by heavy lines) in a data communication session. Start of Block Transfer and End of Block Transfer events are shown by circles on the source user-system interface and destination user-system interface, respectively.

If a given block transfer attempt is not the first in a session and Start of Block Transfer occurs in both the given attempt and the preceding attempt, then the source interblock gap associated with the given attempt is the interval from Start of Block Transfer in the preceding attempt to Start of Block Transfer in the given attempt. Otherwise (if a block transfer attempt is the first in a session, is an attempt in which Start of Block Transfer does not occur, or is preceded by an attempt in which Start of Block Transfer does not occur), the associated source interblock gap is undefined. (Start of Block Transfer does not occur in any attempt that contains only extra BCOs). If a given block transfer attempt is not the first in a session and End of Block Transfer occurs in both the given attempt and the preceding attempt, then the destination interblock gap associated with the given attempt is the interval from End of Block Transfer in the preceding attempt to End of Block Transfer in the given attempt. Otherwise (if a block transfer attempt is the first in a session, is an attempt in which End of Block Transfer does not occur, or is preceded by an attempt in which End of Block Transfer does not occur), the associated destination interblock gap is undefined. (End of Block Transfer does not occur in any attempt that contains only undelivered BCOs).

Each transfer sample selected by **TTSAMP** contains a sequence of consecutive block transfer attempts in a single data communication session and the interblock gaps that precede each block transfer attempt in the sequence. The dashed lines

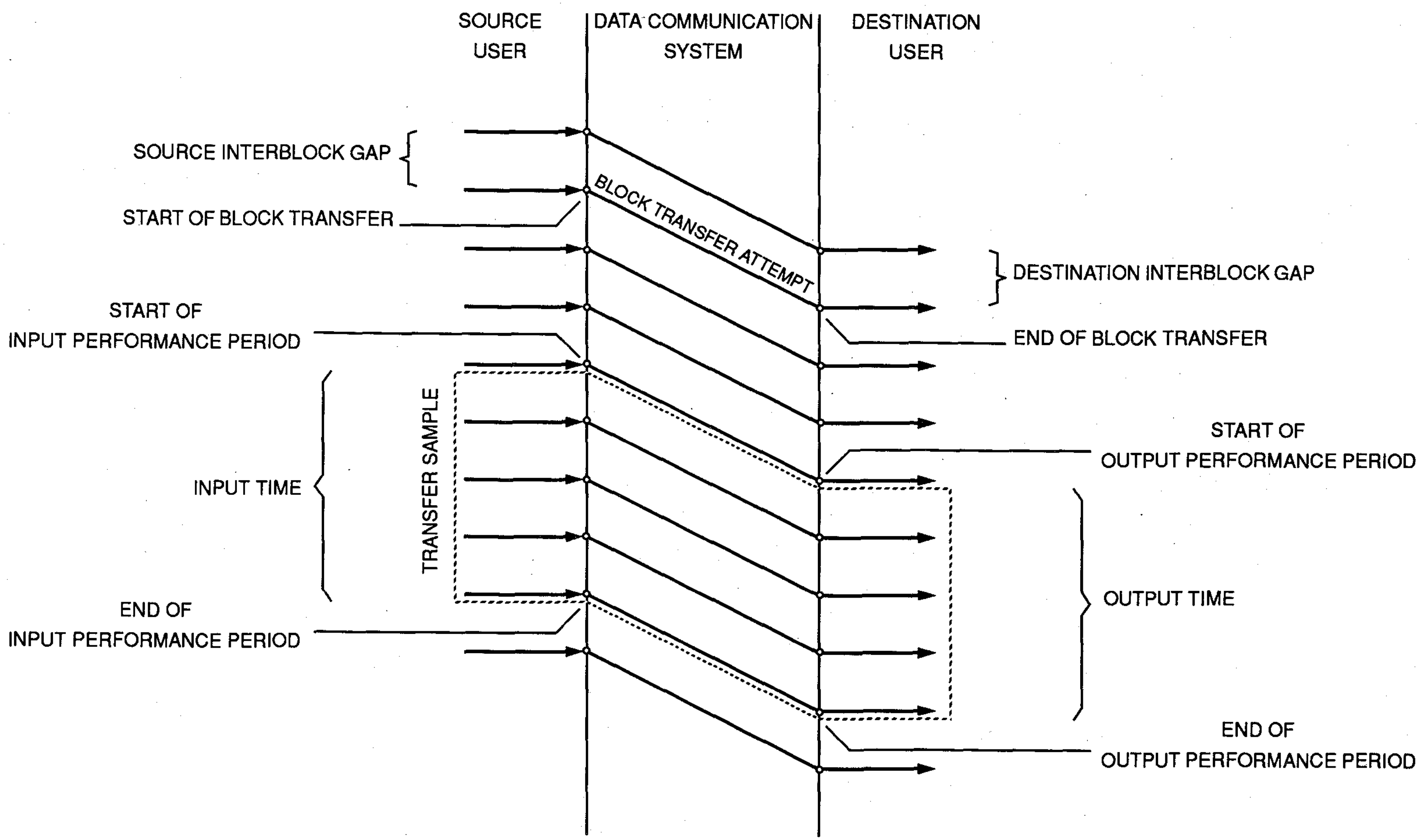


Figure 51. Summary of transfer sample concepts.

in Figure 51 depict a transfer sample containing four successive block transfer attempts.

If Start of Block Transfer occurs in the last attempt that precedes a transfer sample and the last attempt in the sample, then these events delimit the input performance period for the sample. If either delimiting event fails to occur, the input performance period is not defined. When it is defined, the input performance period includes all source interblock gaps associated with block transfer attempts in the sample. If End of Block Transfer occurs in the last attempt that precedes a transfer sample and the last attempt in the sample, then these events delimit the output performance period for the sample. If either delimiting event fails to occur, the output performance period is not defined. When it is defined, the output performance period includes all destination interblock gaps associated with block transfer attempts in the sample.

For a transfer sample, the input time is the duration of the input performance period and the output time is the duration of the output performance period. The input/output time for a transfer sample is the larger of the input and output times for the sample.

4.2.3.2. Selection of Transfer Samples

A block transfer attempt is excluded from any availability or throughput transfer sample selected by **TTSAMP** if

- the attempt is the first in a data communication session or the first in a performance data batch, or
- the outcome of the attempt is Refused Block or is not classified.

If a block transfer attempt is the first in a session, the associated source and destination interblock gaps are not defined. If a block transfer attempt is the first in a batch, the associated interblock gaps are not included in the event history even when defined. A block transfer attempt whose outcome is Refused Block is excluded in order to implement an ANS X3.102 requirement that a transfer sample not be used in the measurement of Transfer Denial Probability if the sample includes any bit transfer attempt in which delivery of a source bit to the destination user is not completed because of user nonperformance. As explained in Section 4.2.2, the outcome of a block transfer attempt is not classified if (i) the attempt contains only undelivered BCOs and (ii) the end of the performance period is a UIT termination event other than a normal UIT termination event generated by a destination entity. A block transfer attempt whose outcome is not classified indicates an anomalous (or erroneous) event history. In the discussion that follows, a block transfer attempt satisfying any of the conditions specified above for exclusion is said to be unacceptable. Otherwise, a block transfer attempt is said to be acceptable.

An availability or throughput transfer sample selected by **TTSAMP** is called complete if the number of bit transfer attempts contained in the sample is equal to or greater than the minimum value given in the specifications input file.

TTSAMP selects the maximum number of complete availability transfer samples from the block transfer attempts recorded in the correlator output file. Each complete availability transfer sample selected by **TTSAMP** therefore contains the smallest number of successive acceptable block transfer attempts for which the total number of bit transfer attempts is equal to or greater than the specified minimum value. An availability transfer sample is terminated when the sample is complete or when **TTSAMP** identifies an unacceptable block transfer attempt. In the former case, the outcome of the transfer sample is determined as described in Section 4.2.3.3. In the latter case, the transfer sample is discarded (it is incomplete) and is not included in the set of performance trials used to measure Transfer Denial Probability.

To ensure that input and output times can be evaluated, a throughput transfer sample must be preceded by and end with block transfer attempts in which both Start of Block Transfer and End of Block Transfer occur. Each throughput transfer sample selected by **TTSAMP** contains the longest sequence of successive acceptable block transfer attempts permitted by these criteria. A throughput transfer sample is terminated only when **TTSAMP** identifies an unacceptable block transfer attempt or when the routine has included the last block transfer attempt in the correlator output file. If a throughput transfer sample terminated by **TTSAMP** is complete, the sample is called a throughput trial and is used to evaluate long-term throughput performance parameters. Otherwise, if a throughput transfer sample terminated by **TTSAMP** is incomplete, the sample is discarded and not used to evaluate long-term throughput performance parameters. **TTSAMP** selects no more than one throughput trial from the block transfer attempts recorded in the correlator output file.

4.2.3.3 Determination of Availability Transfer Sample Outcomes

The performance observed in a complete availability transfer sample is compared with a specified threshold of acceptability for each of four supported user information transfer parameters: Bit Error Probability, Bit Loss Probability, Extra Bit Probability, and User Information Bit Transfer Rate. If the observed performance is equal to or better than the threshold of acceptability for each of the supported parameters, the outcome of the transfer sample is defined to be Successful Transfer. If the observed performance is worse than the threshold of acceptability for one or more supported parameters, the outcome is defined to be Transfer Denial when the failure is attributed to system nonperformance or Rejected Sample when the failure is attributed to user nonperformance. The threshold of acceptability for each supported bit transfer failure probability is defined by ANS X3.102 to be the fourth root of the corresponding specified value given in the specifications input file. The threshold of acceptability for the User Information Bit Transfer Rate is one-third of the corresponding specified value. An availability transfer sample whose outcome is Successful Transfer or Transfer Denial is called a <u>transfer availability trial</u> and is included in the set of performance trials used to estimate Transfer Denial Probability.

Whenever a complete availability transfer sample is obtained, **TTSAMP** determines its outcome in accordance with ANS X3.141. The scheme used by **TTSAMP** is outlined by the flowchart in Figure 52. If the measured value for any of the supported bit transfer failure probabilities exceeds the corresponding threshold value, the outcome is Transfer Denial. If all three measured probabilities are at or below their respective thresholds and if the input and output performance periods for the transfer sample are defined, the routine calculates the measured value of User Information Bit Transfer Rate. If the measured rate is equal to or greater than the threshold rate, the outcome of the transfer sample is Successful Transfer. Otherwise, **TTSAMP** calculates the measured user fraction of input/output time for the sample. If the measured value of the fraction does not exceed the specified value of the fraction, responsibility for the excessive input/output delay is attributed to system nonperformance and the transfer sample

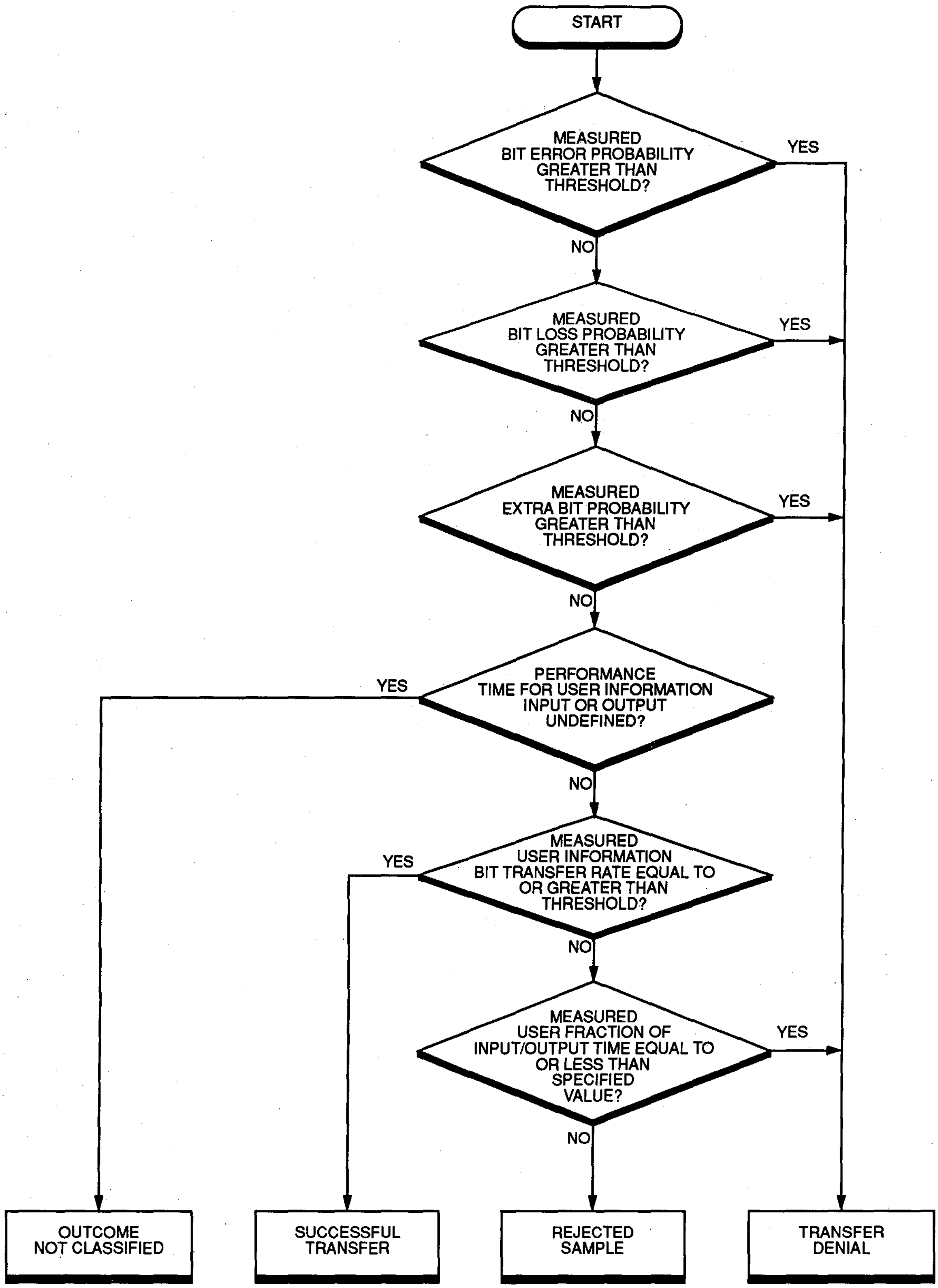


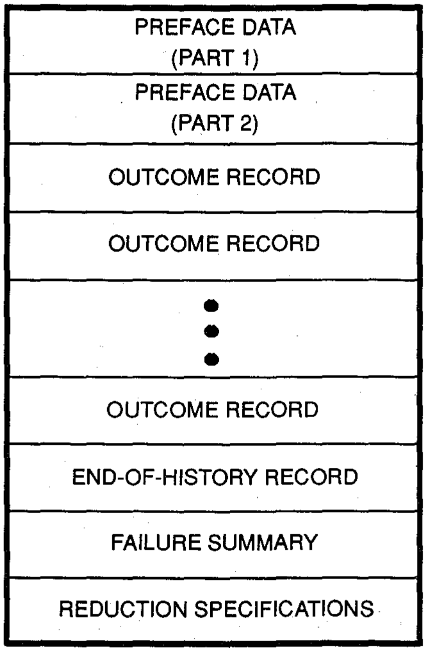
Figure 52. Scheme for determining availability transfer sample outcomes.

outcome is Transfer Denial. If the measured value of the fraction exceeds the specified value, responsibility for the delay is attributed to user nonperformance and the outcome is Rejected Sample. Availability transfer samples whose outcomes are Rejected Sample are excluded from the set of transfer availability trials used to estimate Transfer Denial Probability.

If the input or output performance period for an availability transfer sample is not defined, the input/output time and the User Information Bit Transfer Rate for the sample cannot be determined. If all supported bit transfer failure probabilities for such a sample are at or below the corresponding thresholds, the outcome of the sample is not classified and the sample is excluded from the set of transfer availability trials used to measure Transfer Denial Probability. The input performance period for a transfer sample is not defined whenever Start of Block Transfer does not occur in the last attempt that precedes the sample or in the last attempt in the sample. The output performance period for a transfer sample is not defined whenever End of Block Transfer does not occur in the last attempt that precedes the sample or in the last attempt in the sample.

4.2.3.4 Output Files

Outcomes of all availability transfer samples are recorded in the transfer sample outcome file (**B30**) as they are determined. The record sequence for the file is shown in Figure 53a, record formats are defined in Figure 53b, and an example of a transfer sample outcome file is shown in Figure 53c. The file includes an outcome record for each (nonempty) availability transfer sample selected by **TTSAMP**. This record contains an availability transfer sample outcome code and indexes of the first and last block transfer attempts in the sample. Successful Transfer, Transfer Denial, and Rejected Sample outcomes are indicated by the codes 0, -2, and -5, respectively. Discarded (incomplete) samples are indicated by -8 and complete samples whose outcomes are not classified are indicated by -9. Indexes of the first and last block transfer attempts in an availability transfer sample enable one to readily identify the sample in the correlator output and block transfer outcome files. Various transfer sample descriptors (e.g., numbers of bit transfer attempts in each outcome category and User Information Bit Transfer Rate) can then be evaluated if such data are desired.



a. Record Sequence

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
PREFACE DATA (PART 1):		
1-32	A32	FILE DESCRIPTOR
PREFACE DATA (PART 2):		
1-64	A64	BATCH IDENTIFIER
OUTCOME RECORD:		
1-8	F8.0	AVAILABILITY TRANSFER SAMPLE OUTCOME CODE
9-16	F8.0	INDEX OF FIRST BLOCK TRANSFER ATTEMPT IN AVAILABILITY TRANSFER SAMPLE
17-24	F8.0	INDEX OF LAST BLOCK TRANSFER ATTEMPT IN AVAILABILITY TRANSFER SAMPLE
END-OF-HISTORY RECORD:		
1-8	F8.0	END-OF-HISTORY CODE (-30)
9-16	F8.0	END-OF-HISTORY CODE (-30)
17-24	F8.0	END-OF-HISTORY CODE (-30)
FAILURE SUMMARY:		
1-8	F8.0	NUMBER OF TRANSFER AVAILABILITY TRIALS IN TRANSFER DENIAL MEASUREMENT
9-16	F8.0	NUMBER OF 'TRANSFER DENIAL' OUTCOMES
17-24	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'TRANSFER DENIAL' OUTCOMES
REDUCTION SPECIFICATIONS:		
1-16	F16.8	SPECIFIED VALUE OF USER INFORMATION BIT TRANSFER RATE FOR A TRANSFER AVAILABILITY TRIAL(BITS/SEC)
17-32	F16.8	SPECIFIED VALUE OF USER FRACTION OF INPUT/OUTPUT TIME FOR A TRANSFER AVAILABILITY TRIAL
33-48	F16.8	SPECIFIED VALUE OF BIT ERROR PROBABILITY FOR A TRANSFER AVAILABILITY TRIAL
49-64	F16.8	SPECIFIED VALUE OF BIT LOSS PROBABILITY FOR A TRANSFER AVAILABILITY TRIAL
65-80	F16.8	SPECIFIED VALUE OF EXTRA BIT PROBABILITY FOR A TRANSFER AVAILABILITY TRIAL

b. Record Formats

```
TRANSFER SAMPLE OUTCOME
NTIA-ITS (Boulder)                                          1424
       0.        2.       31.
       0.       32.       61.
      -8.       62.       80.
     -30.      -30.      -30.
       2.        0.        0.
  0.10000000E+05  0.50000000E+00  0.99999999E-08  0.99999999E-08  0.99999999E-08
```

c. Example File

Figure 53. Transfer sample outcome file.

The outcome record corresponding to the final availability transfer sample selected by **TTSAMP** is followed by an end-of-history record. The latter contains an end-of-history code (-30) and has the same format as an outcome record. The subsequent failure summary record lists the total number of transfer availability trials in the Transfer Denial measurement sample and the observed numbers of Transfer Denial outcomes and pairs of consecutive Transfer Denial outcomes. The final reduction specifications record in the transfer sample outcome file lists the specified values of user information transfer performance parameters used to determine outcomes of availability transfer samples.

If a throughput trial (a complete throughput transfer sample) is identified, **TTSAMP** evaluates

- total performance time in the input/output period for the trial (the larger of the input time and the output time) and
- user performance time in the input/output period (the larger of the user performance times in the input period and the output period).

A set of trial descriptors is then recorded in the throughput sample outcome file (**B40**). The record sequence for this file is shown in Figure 54a, record formats are defined in Figure 54b, and an example of a throughput sample outcome file is shown in Figure 54c. The sample range record contains indexes of the first and last block transfer attempts in the throughput trial; these enable one to identify the trial in the correlator output and block transfer outcome files. The final throughput summary record contains observed values of total and user performance times in the input/output period and the number of Successful Bit Transfer outcomes in the throughput trial. This information is used by program **STAR** to estimate values of throughput parameters (User Information Bit Transfer Rate and User Fraction of Input/Output Time) and their confidence limits based on multiple tests (see Volume 5 of this report).

4.3 Disengagement Performance Assessment

In a normal reduction run for which disengagement performance assessment is enabled, procedures in subroutine **DISENG** identify disengagement attempts associated with successful access attempts recorded in a performance data batch and determine their outcomes. Source and destination disengagement attempts are segregated in separate measurement samples.

PREFACE DATA (PART 1)
PREFACE DATA (PART 2)
SAMPLE RANGE RECORD[1]
THROUGHPUT SUMMARY RECORD[1]

[1] Record is included only if throughput transfer sample is complete.

a. Record Sequence

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
PREFACE DATA (PART 1):		
1-32	A32	FILE DESCRIPTOR
PREFACE DATA (PART 2):		
1-64	A64	BATCH IDENTIFIER
SAMPLE RANGE RECORD:		
1-8	F8.0	INDEX OF FIRST BLOCK TRANSFER ATTEMPT IN THROUGHPUT TRANSFER SAMPLE
9-16	F8.0	INDEX OF LAST BLOCK TRANSFER ATTEMPT IN THROUGHPUT TRANSFER SAMPLE
THROUGHPUT SUMMARY RECORD:		
1-8	F8.3	TOTAL PERFORMANCE TIME FOR USER INFORMATION INPUT/OUTPUT IN THROUGHPUT TRANSFER SAMPLE (SECONDS)
9-16	F8.3	USER PERFORMANCE TIME FOR USER INFORMATION INPUT/OUTPUT IN THROUGHPUT TRANSFER SAMPLE (SECONDS)
17-24	F8.0	NUMBER OF 'SUCCESSFUL BIT TRANSFER' OUTCOMES IN THROUGHPUT TRANSFER SAMPLE

b. Record Formats

```
THROUGHPUT SAMPLE OUTCOME
NTIA-ITS (Boulder)                                            1424
      2.      80.
  19.297   4.702  80864.
```

c. Example File

Figure 54. Throughput sample outcome file.

Input to these procedures consists of

- the event history in the consolidated overhead information file (**COI**),
- the event history in the source user information file (**SUI**),
- specified values of Disengagement Time and User Fraction of Disengagement Time used to determine outcomes of source disengagement attempts, specified values of Disengagement Time and User Fraction of Disengagement Time used to determine outcomes of destination disengagement attempts, and the specified value of Access Time used to identify successful access attempts.

All specified values indicated above are obtained from the consolidated specifications file (**CSP**).

Outcomes of individual source disengagement attempts are recorded in the source disengagement outcome file (**D1O**), and outcomes of individual destination disengagement attempts are recorded in the destination disengagement outcome file (**D2O**).

Procedures for identifying disengagement attempts are discussed in Section 4.3.1 and procedures for determining their outcomes are discussed in Section 4.3.2.

4.3.1 Identification of Disengagement Attempts

Subroutine **DISENG** identifies the start of a disengagement attempt and the end of the associated performance period. The start of a disengagement attempt always corresponds to a Disengagement Request event. In a connection-oriented session, the disengagement functions for both users start with a single Disengagement Request at one of the user interfaces. In a connectionless session, disengagement functions for the two users start with separate Disengagement Requests at the respective user interfaces. A Disengagement Request at a given interface is represented in the consolidated overhead information file by an event record in which the local user and adjacent half-system undergo transitions from a committed (2 or 3) state to a closing (4 or 5) state. A record of a Disengagement Request at the source interface in a connection-oriented session is illustrated in Figure 14.

The end of a disengagement performance period corresponds to whichever occurs first: the end of the disengagement attempt or the end of the maximum performance period (the performance deadline) associated with the attempt. In accordance with ANS X3.102, the length of the maximum performance period for a disengagement attempt is three times the applicable specified value of Disengagement Time. Disengagement timeout occurs if the end of a disengagement attempt does not occur on or before the associated performance deadline.

The end of a disengagement attempt normally corresponds to a Disengagement Confirmation signal at the interface between the disengaging user and the system. This event is represented in the consolidated overhead information file by an event record in which the disengaging user and the adjacent half-system undergo transitions from a closing (4 or 5) state to an idle (0 or 1) state. Disengagement Confirmation signals for both source and destination disengagement attempts are illustrated in Figure 14.

Subsequent to a Disengagement Request at a given interface, both local entities are in the closing state. Any event in which either a disengaging user or the adjacent half-system undergoes a transition from a closing state is regarded by **DISENG** as the end of the disengagement attempt.

The first disengagement attempt in a connection-oriented session refers to the disengagement of the user at the interface where the Disengagement Request occurs (the local interface); the second disengagement attempt refers to the disengagement of the user at the other interface (the remote interface). The first disengagement attempt in a connectionless session refers to the disengagement of the user at the interface where the initial Disengagement Request occurs; the second disengagement attempt refers to the disengagement of the user at the interface where the second Disengagement Request occurs. In the unlikely case that Disengagement Requests for source and destination users in a connectionless session have the same event time, the request at the source interface appears first in the consolidated event history. Because the procedure for identifying the first disengagement attempt in a session differs significantly from the procedure for identifying the second disengagement attempt, the two procedures are discussed separately.

4.3.1.1 Identification of First Disengagement Attempt in Session

DISENG observes the earliest UIT termination event (see Section 4.2.2) that is coincident with or later than the start of user information transfer in the session. If the observed UIT termination event is normal, it is a Disengagement

Request and is the start of the first disengagement attempt in the session. If the observed UIT termination event is anomalous (e.g., a transition to an idle state by some entity), the event history is anomalous (or erroneous) and **DISENG** does not identify any disengagement attempt in the session.

In performance assessment procedures implemented by **DISENG**, the end of the performance period for the first disengagement attempt in a session corresponds to whichever occurs first after the Disengagement Request:

- a transition from a closing state by either disengaging entity,
- the associated performance deadline, or
- an Access Request for a subsequent session.

If the performance period for the first disengagement attempt in a session is terminated by an Access Request for a subsequent session, the event history is anomalous (or erroneous) and **DISENG** does not classify the outcome of the disengagement attempt. An Access Request for a subsequent session can terminate the performance period for the first disengagement attempt only if the disengaging user in that attempt is the nonoriginating user.

4.3.1.2 Identification of Second Disengagement Attempt in Session

In a connection-oriented session, the start of the second disengagement attempt corresponds to the Disengagement Request for the first attempt. In a connectionless session (where the start of the second disengagement attempt corresponds to a separate Disengagement Request), **DISENG** observes whichever occurs first after the Disengagement Request for the first attempt in the session:

- a UIT termination event at the relevant interface, or
- an Access Request for a subsequent session.

If **DISENG** observes a normal UIT termination event, it is a Disengagement Request and is the start of the second disengagement attempt in the session. If **DISENG** observes an anomalous UIT termination event or an Access Request for a subsequent session (the latter event is observed only if the disengaging user in the first attempt is the originating user), the event history is anomalous (or erroneous); in this case, **DISENG** does not identify a second disengagement attempt in the session. The relevant interface in the second disengagement attempt in a session is the interface between the disengaging user and the system.

In performance assessment procedures implemented by **DISENG**, the end of the performance period for the second disengagement attempt in a session corresponds to whichever occurs first after the Disengagement Request:

- a transition from a closing state by either disengaging entity,
- the associated performance deadline,
- an Access Request for a subsequent session, or
- an anomalous UIT termination event at the relevant interface (in a connection-oriented session).

If the performance period for the second disengagement attempt in a session is terminated by an Access Request for a subsequent session or by an anomalous UIT termination event, the event history is anomalous (or erroneous) and **DISENG** does not classify the outcome of the disengagement attempt. An Access Request for a subsequent session can terminate the performance period for the second disengagement attempt only if the disengaging user in that attempt is the nonoriginating user.

In a connection-oriented session, both entities at the relevant interface in the second disengagement attempt are in a committed state both before and after the Disengagement Request (which occurs at the other interface). The normal event sequence at the relevant interface includes a normal UIT termination event (whose representation in the consolidated overhead information file has the form of a Disengagement Request).

4.3.2 Determination of Disengagement Outcomes

Subroutine **DISENG** determines outcomes of source and destination disengagement attempts in accordance with the definitions given in ANS X3.102. These outcomes may be characterized as follows:

> Successful Disengagement occurs if a Disengagement Confirmation signal for the attempt occurs no later than the disengagement performance deadline.

Disengagement Denial occurs if disengagement timeout occurs and the measured user fraction of performance time for the period does not exceed the specified value of User Fraction of Disengagement Time.

User Disengagement Blocking occurs if disengagement timeout occurs and the measured user fraction of performance time for the period exceeds the specified value of User Fraction of Disengagement Time.

4.3.2.1 Outcome Determination Scheme

The scheme used in subroutine **DISENG** to determine outcomes of disengagement attempts is outlined by the flowchart in Figure 55. Procedures in **DISENG** that identify the end of the performance period for a disengagement attempt assign the attempt to one of the following categories: (i) disengagement attempts in which the performance period is terminated by a transition from a closing state, or (ii) disengagement attempts in which the performance period is terminated by disengagement timeout.

The outcome of a disengagement attempt in which the performance period is terminated by a transition from a closing state depends on the nature of the transition. If the transition corresponds to a Disengagement Confirmation signal, the outcome of the disengagement attempt is Successful Disengagement. If some other transition from a closing state terminates the period, the outcome of the disengagement attempt is not classified (the event history is anomalous or erroneous). Disengagement attempts whose outcomes are not classified are excluded from the set of trials used to estimate values of disengagement performance parameters.

For a disengagement attempt in which the performance period is terminated by disengagement timeout, subroutine **DISENG** evaluates the user fraction of performance time for the period. If the measured fraction exceeds the applicable specified value for User Fraction of Disengagement Time, responsibility for the excessive delay is attributed to user nonperformance and the outcome of the disengagement attempt is User Disengagement Blocking. Otherwise, responsibility for the delay is attributed to system nonperformance and the outcome is Disengagement Denial.

4.3.2.2 Disengagement Outcome Files

Outcomes of individual source or destination disengagement attempts are respectively recorded in the source disengagement outcome file (**D10**) or

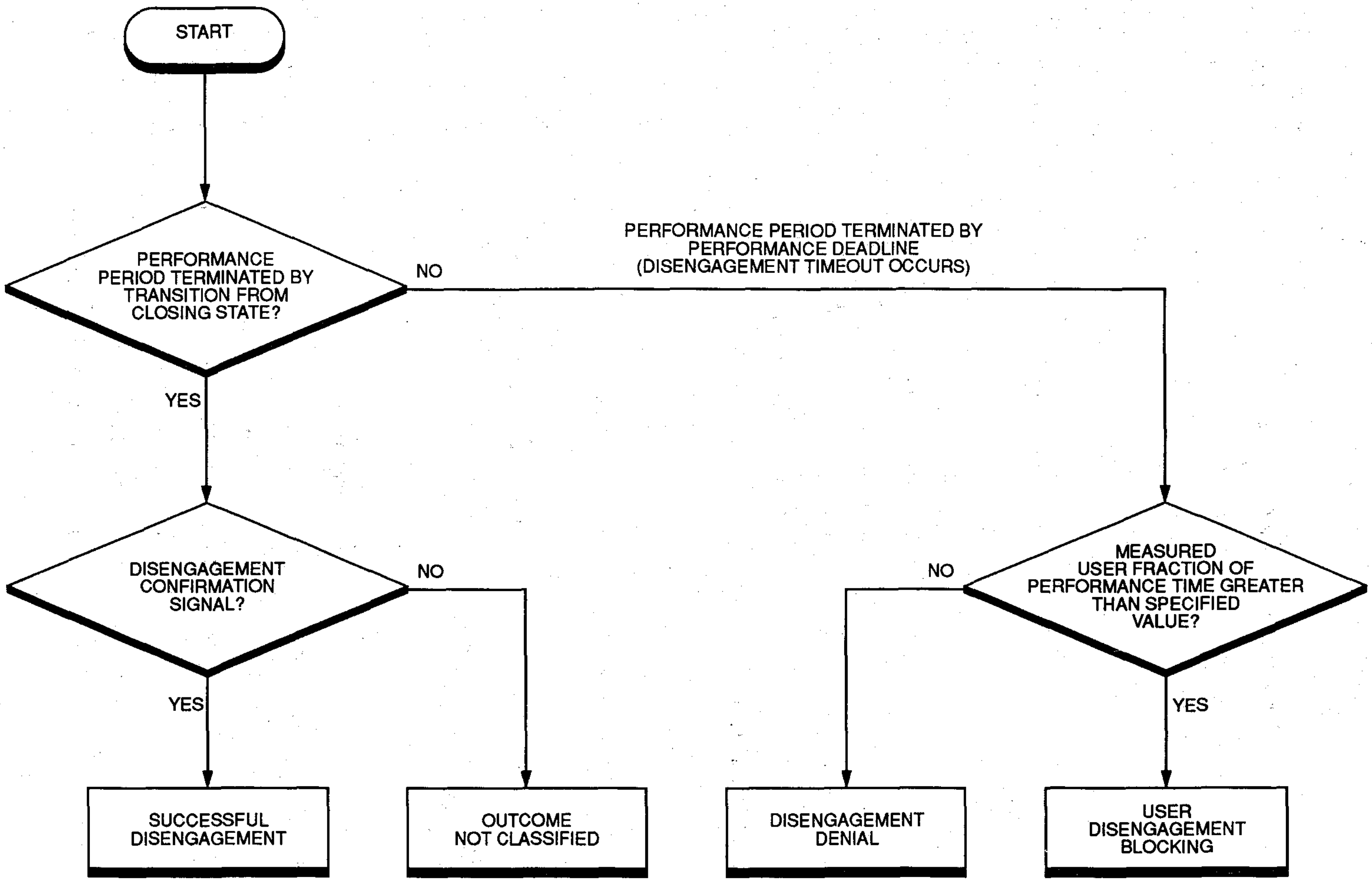


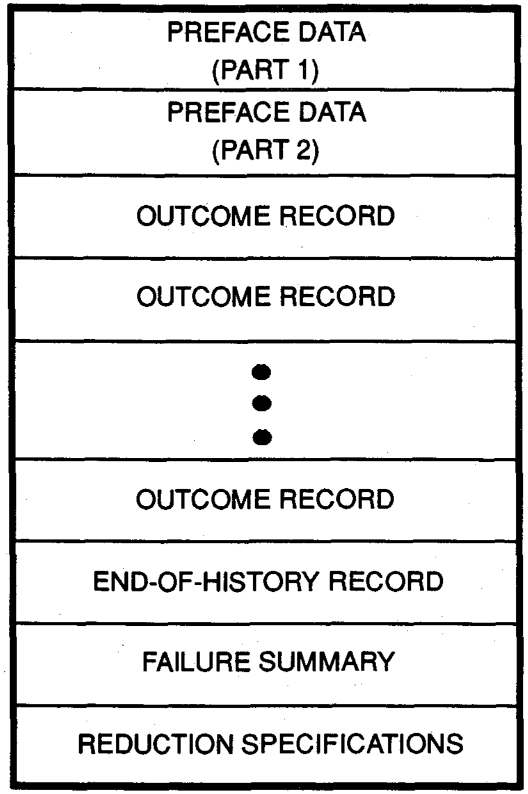
Figure 55. Scheme for determining disengagement outcomes.

destination disengagement outcome file (D2O) as they are determined. The record sequence in these files is shown in Figure 56a, record formats are defined in Figure 56b, and examples of source and destination disengagement outcome files are shown in Figure 57. Each source or destination disengagement attempt identified by subroutine **DISENG** is represented by an <u>outcome record</u> in the respective disengagement outcome file. The outcome record for a successful disengagement attempt contains overall and user performance times, whereas the record for an unsuccessful attempt contains a negative <u>disengagement outcome code</u>. Disengagement Denial and User Disengagement Blocking outcomes are indicated by -2 and -5, respectively. Disengagement attempts whose outcomes are not classified are indicated by -9. The final outcome record in a file is followed by an <u>end-of-history record</u>. The latter contains an <u>end-of-history code</u> (-30) and has the same format as the outcome record for an unsuccessful disengagement attempt. The subsequent <u>failure summary record</u> lists the number trials in the measurement sample and observed numbers of Disengagement Denial outcomes and pairs of consecutive Disengagement Denial outcomes. The final <u>reduction specifications record</u> in a disengagement outcome file contains the specified values of Disengagement Time and User Fraction of Disengagement Time used in outcome determination. This record also contains the specified value of Access Time used to identify successful access attempts. Information in the disengagement outcome files enables the statistical analysis program **STAR** (described in Volume 5 of this report) to calculate estimated values and their confidence limits for all disengagement performance parameters defined in ANS X3.102.

4.4 Performance Time Allocation

During the performance of a particular function, delay attributed to user activities is required to estimate values of ancillary performance parameters and assign responsibility for timeout failures to system or user nonperformance. User performance time for access, block transfer, and disengagement performance periods is evaluated by subroutine **ANCILL**. For user information input and output periods associated with availability and throughput transfer samples, user performance time is evaluated by subroutine **UIOTIM**.

Concepts employed by **ANCILL** and **UIOTIM** are outlined in Figure 58a. The performance period is divided into a sequence of <u>communication state intervals</u>



a. Record Sequence

CHARACTER FIELD	EDIT DESCRIPTOR	CONTENTS
PREFACE DATA (PART 1):		
1-32	A32	FILE DESCRIPTOR
PREFACE DATA (PART 2):		
1-64	A64	BATCH IDENTIFIER
OUTCOME RECORD (SUCCESSFUL PERFORMANCE):		
1-8	F8.3	OVERALL PERFORMANCE TIME FOR DISENGAGEMENT ATTEMPT (SECONDS)
9-16	F8.3	USER PERFORMANCE TIME FOR DISENGAGEMENT ATTEMPT (SECONDS)
OUTCOME RECORD (UNSUCCESSFUL PERFORMANCE):		
1-8	F8.0	OUTCOME CODE FOR UNSUCCESSFUL DISENGAGEMENT ATTEMPT
9-16	F8.0	OUTCOME CODE FOR UNSUCCESSFUL DISENGAGEMENT ATTEMPT
END-OF-HISTORY RECORD:		
1-8	F8.0	END-OF-HISTORY CODE (-30)
9-16	F8.0	END-OF-HISTORY CODE (-30)
FAILURE SUMMARY:		
1-8	F8.0	NUMBER OF DISENGAGEMENT ATTEMPTS IN MEASUREMENT SAMPLE
9-16	F8.0	NUMBER OF 'DISENGAGEMENT DENIAL' OUTCOMES
17-24	F8.0	NUMBER OF PAIRS OF SUCCESSIVE 'DISENGAGEMENT DENIAL' OUTCOMES
REDUCTION SPECIFICATIONS:		
1-16.	E16.8	SPECIFIED VALUE OF DISENGAGEMENT TIME (SECONDS)
17-32	E16.8	SPECIFIED VALUE OF USER FRACTION OF DISENGAGEMENT TIME
33-48	E16.8	SPECIFIED VALUE OF ACCESS TIME (SECONDS)

b. Record Formats

Figure 56. Record sequence and record formats in disengagement outcome files.

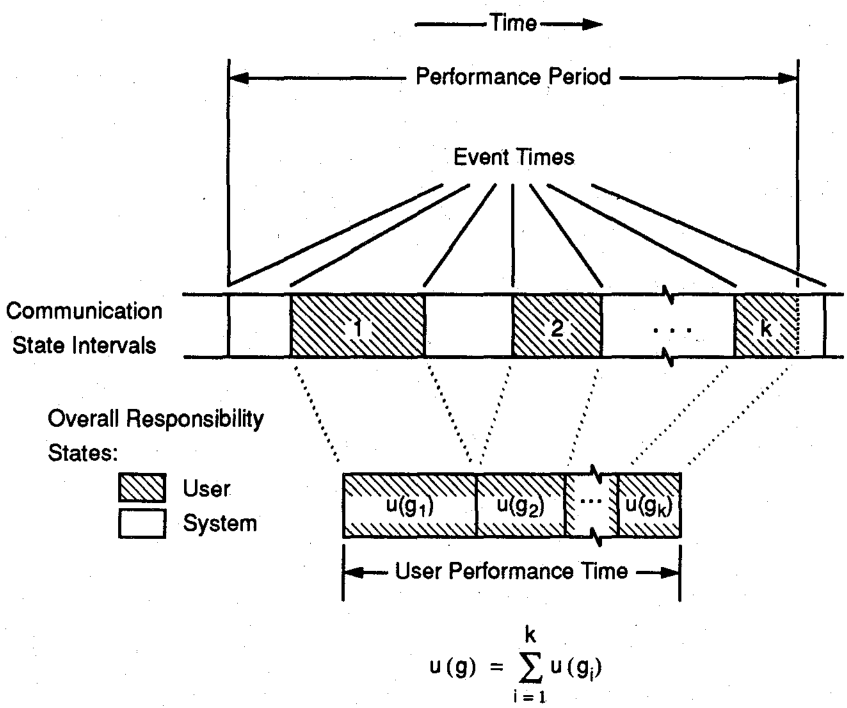
```
SOURCE DISENGAGEMENT OUTCOME
NTIA - PDN test from Washington, DC                              0915
  13.448    0.727
  13.438    0.867
  13.443    0.919
  13.438    0.932
  14.453    0.913
  15.444    0.938
  13.444    0.916
  13.440    0.921
  13.431    0.915
  18.453    0.929
     -2.       -2.
  15.483    0.928
  13.238    0.931
  13.434    0.889
  13.446    0.927
  13.447    0.879
  13.441    0.882
    -30.      -30.
     17.        1.       0.
  0.14000000E+02  0.75000003E-01  0.45000000E+02
```

a. Source Disengagement Outcome File

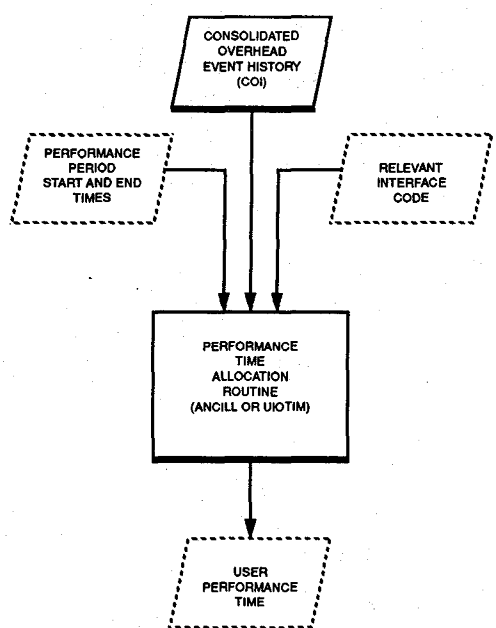
```
DESTINATIONDISENGAGEMENT OUTCOME
NTIA - PDN test from Washington, DC                              0915
    3.705   0.461
    3.626   0.601
    3.605   0.653
    3.583   0.666
    3.672   0.647
    3.904   0.672
    3.649   0.650
    3.731   0.655
    3.597   0.649
    4.239   0.663
    3.632   0.669
    4.286   0.662
    3.716   0.665
    3.607   0.623
    3.681   0.661
    3.604   0.613
    3.570   0.616
    -30.      -30.
     17.        0.       0.
  0.40000000E+01  0.20000000E+00  0.45000000E+02
```

b. Destination Disengagement Outcome File

Figure 57. Examples of source and destination disengagement outcome files.



a. Allocation Concepts



b. Input/Output Summary

Figure 58. Outline of performance time allocation.

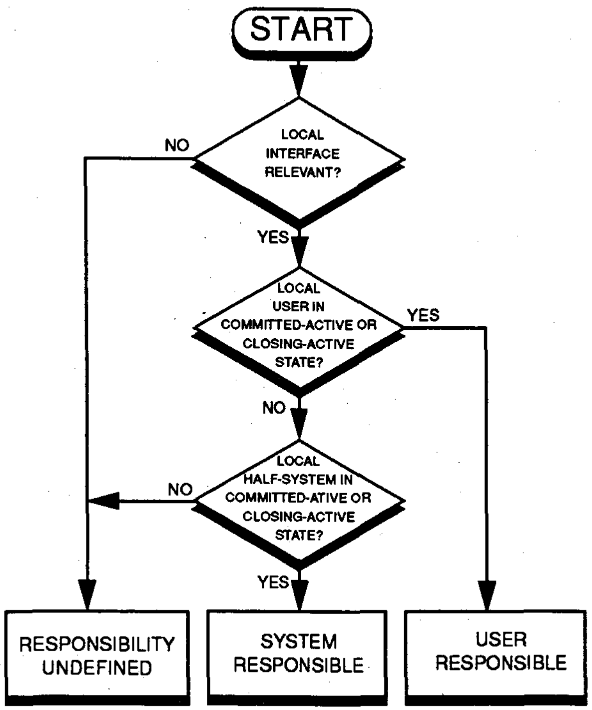
by recorded overhead events (communication state transitions) included in the period. On the basis of local responsibility states (ancillary communication states) at the monitored source and destination interfaces, the allocation routine assigns an overall responsibility state to each interval: user responsible or system responsible. User performance time for the period is the sum of the durations of intervals of overall user responsibility. An outline of the allocation process is shown in Figure 58b. Input to the process consists of

- the event history in the consolidated overhead information file (COI),
- start and end times for the performance period, and
- a code specifying the interfaces that are relevant in allocating performance time for the given period.

The allocation routine (**ANCILL** or **UIOTIM**) examines the event history (which lists, for each interval, the communication states of the user and system at both monitored interfaces) to identify intervals of overall user responsibility within the performance period and evaluates the user performance time for the period.

Local responsibility states for an interval depend on the relevance of the local interface for performance time allocation and (for a relevant interface) on the communication states of the local entities. These states - user responsible, system responsible, and responsibility undefined - are determined according to the scheme shown in Figure 59a. Relevant interfaces for evaluating user performance time are shown in Figure 59b for each type of performance period. For access, block transfer, and disengagement performance periods, the relevant interfaces are the same for all communication state intervals in the period. For input and output performance periods associated with a transfer or throughput sample, the relevant interface may differ from one interval to another.

The overall responsibility state for an interval is jointly determined by the local responsibility states at the source and destination interfaces as specified in ANS X3.141 and shown in Figure 59c. The matrix includes a pair of split responsibility states in which the user is responsible at one interface and the system is responsible at the other. Intervals of split responsibility are accounted for in performance time allocation by including them in the earliest subsequent interval of overall user or system responsibility. If a user and the



a. Local Responsibility State

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

b. Relevant Interfaces

Local Responsibility State at Source Interface

Local Responsibility State at Destination Interface	USER RESPONSIBLE	SYSTEM RESPONSIBLE	RESPONSIBILITY UNDEFINED
USER RESPONSIBLE	USER RESPONSIBLE	"SPLIT" RESPONSIBILITY	USER RESPONSIBLE
SYSTEM RESPONSIBLE	"SPLIT" RESPONSIBILITY	SYSTEM RESPONSIBLE	SYSTEM RESPONSIBLE
RESPONSIBILITY UNDEFINED	USER RESPONSIBLE	SYSTEM RESPONSIBLE	

Overall Responsibility States

c. Overall Responsibility States

Figure 59. Local and overall responsibility states.

system simultaneously delay completion of a function, responsibility for the joint delay is thus attributed to whichever entity delays longer. Local and overall responsibility states for a communication state interval are determined by a separate subroutine (**ORSTAT**).

5. PRODUCTION OF PERFORMANCE ASSESSMENT SUMMARY

In the final phase of a normal reduction run, I/O routines produce a user-oriented summary of assessment results. These results are written to the assessment summary file (**SUM**). Production procedures are organized as a hierarchy of processes shown by the block diagram in Figure 60. The overall process is implemented by program **EPILOG** and subordinate processes are implemented by the subroutines indicated in the diagram. General features of these processes are outlined in the paragraphs that follow.

Subroutine **ELIPUT** implements the input of data to be included in the assessment summary. Subroutine **RDSPEC** reads reduction specifications from the consolidated specifications file, subroutine **RDSTAT** reads performance statistics from the statistics file, and subroutine **RDPARS** reads measured performance parameter values from the parameters file. If user information transfer performance assessment is enabled in the run, subroutine **RDCSUM** reads a summary of correlation results from the correlator output file.

Subroutine **SUMMRY** calls the subroutines indicated in the diagram to write particular portions of the assessment summary file. **ASPREF** writes the assessment summary preface, which lists various descriptors pertaining to the underlying performance measurement test. An example of an assessment summary preface is illustrated in Figure 61. The batch identifier described in Section 3.1 is displayed between the dashed lines. As outlined in Section 3.3, the measurement start and end times correspond, respectively, to the earliest and latest events contained in the source and destination overhead event histories included in the performance data batch. These events are identified by subroutine **CKINFO** during the preliminary data examination phase of a reduction run. The event times are converted by subroutine **DATTIM** from seconds after the specified reference time (included in preface data in the overhead information files) to the local date and time-of-day at the originating user site.

If access performance assessment is enabled in the reduction run, subroutine **ASUMRY** writes the access assessment summary. An example of an access assessment summary is illustrated in Figure 62. This summary lists performance statistics (observed outcome counts), measured parameter values, and the specified values of Access Time and User Fraction of Access Time used in determining outcomes of access attempts.

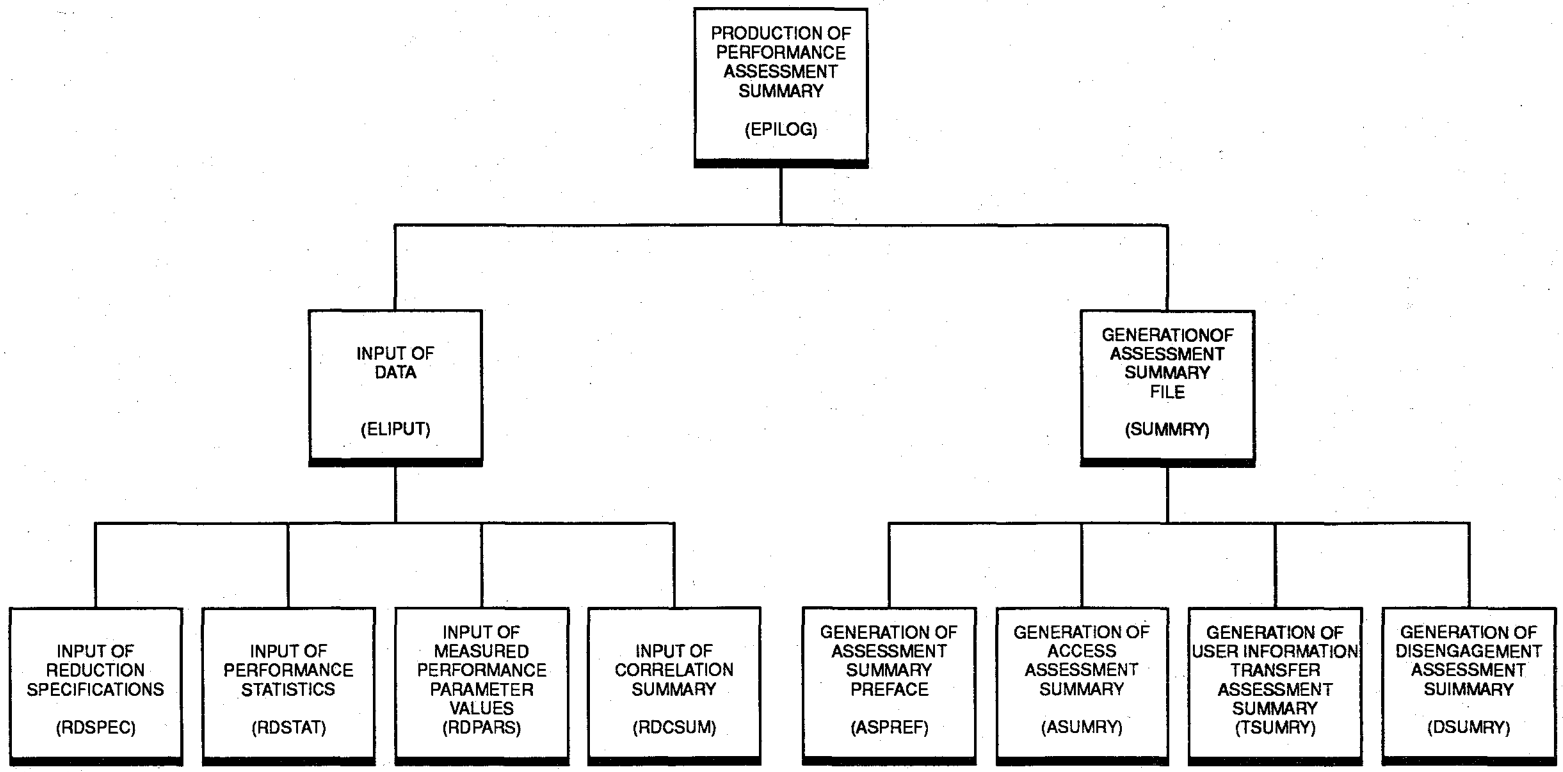


Figure 60. Organization of **EPILOG** processes.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*                                                                             *
*                   PERFORMANCE ASSESSMENT  SUMMARY                           *
*                                                                             *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *


* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*                                                                             *
*                       PERFORMANCE DATA  BATCH                               *
*                                                                             *
*     ------------------------------------------------------------------      *
*     NTIA - PDN test from Washington, DC                          0915       *
*     ------------------------------------------------------------------      *
*                                                                             *
*                                                                             *
*     SOURCE USER . . . . . . . . . . NTIA - Term2 (NBS-Gaithersburg)         *
*                                                                             *
*     DESTINATION USER . . . . . . . . NTIA - Host1 (Boulder)                 *
*                                                                             *
*     ORIGINATING USER . . . . . . . . . . . . . . . . . . . . . . SOURCE     *
*                                                                             *
*                                                                             *
*     SESSION CATEGORY . . . . . . . . . . . . . .  CONNECTION ORIENTED       *
*                                                                             *
*     INITIAL DISENGAGEMENT CATEGORY . . . . . . . . . . . . NEGOTIATED       *
*                                                                             *
*                                                                             *
*     MEASUREMENT START TIME . . . . . . . .  12/12/83  16:10:59.389 LT       *
*                                                                             *
*     MEASUREMENT END TIME . . . . . . . . .  12/12/83  16:51:11.027 LT       *
*                                                                             *
*     (LT - LOCAL TIME-OF-DAY AT ORIGINATING USER SITE)                       *
*                                                                             *
*                                                                             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

Figure 61. Example of assessment summary preface.

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

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

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                            *
*                         ----------------------                             *
*                         PERFORMANCE  STATISTICS                            *
*                         ----------------------                             *
*                                                                            *
*  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 62. Example of access assessment summary.

If user information transfer performance assessment is enabled in the reduction run, subroutine **TSUMRY** writes the user information transfer assessment summary. An example of this summary is illustrated in Figure 63. Part 1 of the summary lists correlation results and specified values of constants used in the correlation process. Part 2 lists performance statistics and Part 3 lists measured values of user information transfer parameters and the specified parameter values used in determining outcomes of block transfer attempts and availability transfer samples.

If disengagement performance assessment is enabled in the reduction run, subroutine **DSUMRY** writes the source disengagement summary and the destination disengagement summary. Examples of these disengagement summaries are shown in Figures 64 and 65. Each summary lists performance statistics, measured parameter values, and specified parameter values used in determining disengagement outcomes.

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

* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                           *
*              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 63 (Part 1). Example of user information transfer assessment summary.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*              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 63 (Part 2). Example of user information transfer assessment summary.

```
------------------------------------------------------------------
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 63 (Part 3). Example of user information transfer assessment summary.

```
------------------------------------------------------------------
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 64. Example of source disengagement assessment summary.

```
---------------------------------------------------------------
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 65. Example of destination disengagement assessment summary.

6. ACKNOWLEDGEMENTS

Several members of the ITS staff contributed to the development of the NTIA data reduction software and the preparation of Volume 4 of this report. David Wortendyke, Chris Bogart, and Darin Schwartz wrote the initial version of the shell script implementation of the reduction programs; this was ably revised by Margaret Pinson. D. J. Atkinson expertly utilized a computer graphics system to convert the author's many sketches to professional-quality drawings. M. J. Miles provided many helpful comments during the writing of this volume. Carole Ax prepared the final manuscript. Timothy Butler and Marjorie Weibel served as technical reviewers. To all of these contributors, the author expresses his sincere thanks.

7. REFERENCES

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 (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).

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 evaluation of data communication services: measurement design, conduct, and results, NTIA Report 88-238, August, 294 pp. (NTIS Order Number PB 89-117519/AS).

Wortendyke, D.R., N.B. Seitz, K.P. Spies, E.L. Crow, and D.S. Grubb (1982), User-oriented performance measurements on the ARPANET: the testing of a proposed Federal Standard, NTIA Report 82-112, November, 293 pp. (NTIS Order Number PB 83-159947).

APPENDIX: SHELL SCRIPT IMPLEMENTATION OF REDUCTION PROGRAMS

In the NTIA implementation of American National Standard X3.141, data collected in a single performance measurement test are processed by an operator-invoked shell script **do**. The data reduction portion of the processing is outlined in Figure A-1 by a structured design diagram.

For an access-disengagement test whose test number is represented by nnnn, data reduction is accomplished by a shell script **reduc-a** (invoked by **do**). This shell script first copies the file **spi.acd** into the file **SPI**, and copies files **nnnn.soi**, **nnnn.doi**, and **nnnn.sui** into files **SOI**, **DOI**, **and SUI**, respectively, for input to the reduction programs. The file **spi.acd** is created prior to running **do**, as described in Section 5 of Volume 2 of this report; the files **nnnn.soi**, **nnnn.doi**, and **nnnn.sui** are generated earlier by the off-line data extraction (data conversion) process in **do**. A C program (**batchid**) next extracts the identifier for the current batch from **SOI** and writes it to **SPI**. The shell script **reduc-a** then calls **prolog**, **analyz**, and **epilog** to carry out the functions indicated in the figure and outlined in Section 2 of this volume. The outcome files **ACO**, **D1O**, and **D2O** output by **analyz** serve as input to the subsequent data analysis procedures in **do**; these are described in Volume 5 of this report.

For a user information transfer test, the analogous data reduction procedures are accomplished by the shell script **reduc-x**. This shell script copies the file **spi.xfr** (created prior to running **do**, as described in Section 5 of Volume 2 of this report) into **SPI**, and copies files **nnnn.soi**, **nnnn.doi**, **nnnn.sui**, and **nnnn.dui** (generated earlier by the off-line data extraction (data conversion) process in **do**) into **SOI**, **DOI**, **SUI**, and **DUI**, respectively, for input to the reduction programs. After **batchid** extracts the current batch identifier from **SOI** and writes it to **SPI**, the shell script **reduc-x** calls **prolog**, **analyze**, and **epilog** to carry out the functions indicated in Figure A-1. The outcome files **B1O**, **B2O**, **B3O**, and **B4O** output by **analyz** serve as input to subsequent data analysis procedures in **do**, as described in Volume 5 of this report.

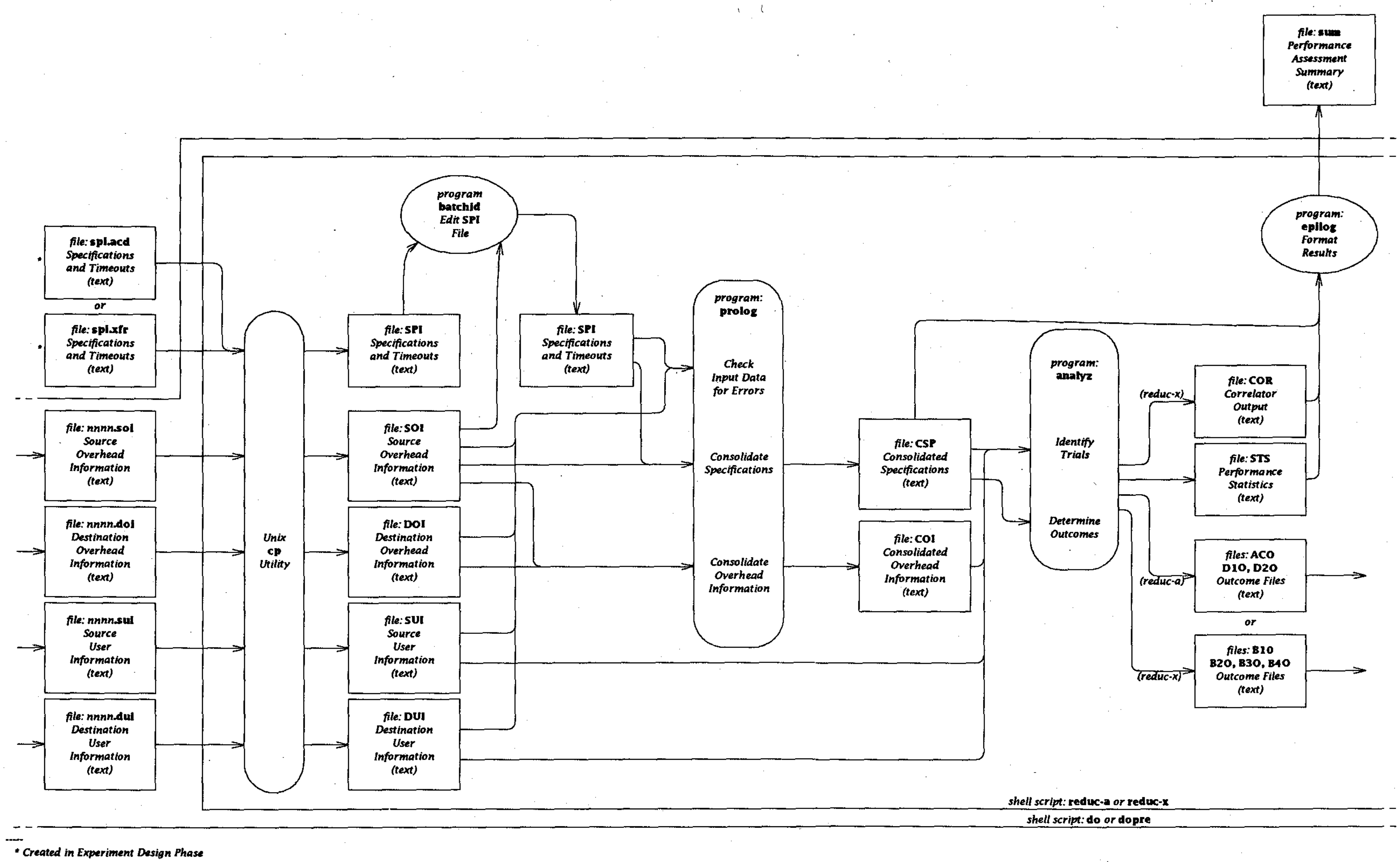


Figure A-1. Structured design diagram of data reduction process.

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 (4)		

4. TITLE AND SUBTITLE
Performance Evaluation of Data Communication Services: NTIA Implementation of American National Standard X3.141, Volume 4. Data Reduction

5. Publication Date
August 1995

6. Performing Organization Code
NTIA/ITS.N3

7. AUTHOR(S)
Kenneth P. Spies

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 shows how the data reduction phase is implemented by a set of FORTRAN computer programs and associated I/O files. These programs examine extracted performance data to identify individual trials and determine their outcomes. This volume outlines the data reduction process, describes input to the process, and discusses key concepts used by data reduction procedures.

16. Key Words *(Alphabetical order, separated by semicolons)*

American National Standards; data communications; end users; performance measurements; performance parameters

17. AVAILABILITY STATEMENT	18. Security Class. *(This report)*	20. Number of pages
☒ UNLIMITED.	Unclassified	156
☐ FOR OFFICIAL DISTRIBUTION.	19. Security Class. *(This page)* Unclassified	21. Price:

*U.S. GOVERNMENT PRINTING OFFICE: 1995-676-490/25108