



Government Systems



1N-18-CR  
2-2-90  
P. 70

# Simulator Design

for  
Advanced ISDN Satellite Design  
and Experiments

30 March 1992

(NASA-CR-190140) SIMULATOR DESIGN FOR  
ADVANCED ISDN SATELLITE DESIGN AND  
EXPERIMENTS Task Completion Report, 25 Sep.  
1990 - 1 Mar. 1992 (GTE Government Systems  
Corp.) 70 p

N92-20321

Uncl 16  
0079290

CSCC 22R 05/18

Task Completion Report  
NASA SCAR Contract NASW-4520, 13 Sep 1990

Prepared by  
Gerard R. Pepin  
GTE/Government Systems  
15000 Conference Center Drive  
Chantilly, Virginia 22021-3808

**SECTION 1**  
**INTRODUCTION**  
**SECTION 1**  
**INTRODUCTION**

**1.1 Background**

The objectives of this element of the NASA Satellite Communications Applications Research (SCAR) Program are to develop new advanced on-board satellite capabilities that will enable the provision of new services, namely interim and full Integrated Services Digital Network (ISDN) services via satellite and to provide a system analysis of futuristic satellite communications concepts, namely broadband services via satellite.

This aspect of the NASA SCAR Program provides a research and development effort to:

- 1) develop basic technologies and concepts to use the on-board processing and switching capabilities of advanced satellites that will enable the provision of interim and full ISDN services and
- 2) provide a systems and requirements analysis of future satellite communications concepts based on a new generation of broadband switching and processing satellites.

These objectives will be achieved in part via modeling and simulation of ISDN satellites as part of the ISDN terrestrial network. Models of the Interim Service ISDN Satellite (ISIS) and the Full Service ISDN Satellite (FSIS) described in their Task Completion Report dated 15 Sep 1991 and 1 Mar 92, respectively, will be exercised using discrete event simulation techniques. To provide meaningful results these network models represent the subsystems of the advanced satellite system at a proper level of abstraction to include real world ISDN communications satellite design parameters.

An end-to-end network view was developed using the framework of the CCITT and ANSI standards to ensure that ISDN procedures and protocols are properly implemented to permit meaningful simulation, evaluation and analyses of ISDN communications satellite designs. Performance measures and scenarios published in a recent update report, dated 28 Feb 1992, addressed simulation results in terms of throughput, response time, blocking probability, and robustness. Simulation measurements must provide insight into the engineering viability of the ISDN communication satellite systems by associating them with parameters such as propagation delay, signal degradation, message queue lengths, network node switching delays and protocol timers.

**1.2 Scope**

This task completion report documents the simulation techniques that will be associated with the network model associated with both the ISIS and FSIS architectures. The approach is described in Figure 1.1-1, "NASA/SCAR Approaches for Advanced ISDN Satellites". The ISIS Network Model design represents satellite systems like the Advanced Communications Technology Satellite (ACTS) orbiting switch.



Government  
Systems



### ISIS Interim Service ISDN Satellite

- ACTS-like Satellite Design and Transponder
- Provide Narrowband ISDN Services (Basic Rate Access)
- Provide remote access ISDN Satellite Terminals using ISDN Satellite Terminal Adapter
- Will use D channel signaling but NOT SS7
- Will use ACTS call control and Baseband Switching Architecture

### FSIS Full Service ISDN Satellite

- New ISDN Satellite Design with onboard Class 5 Switch and SS7 Network Interface
- Provide Narrowband ISDN Services (Basic/Primary Rate Access)
- Provide nationwide single hop single CONUS earth coverage antenna satellite link connectivity to an interexchange node for ISDN Satellite Terminals (up to 10,000 ISAT)
- Will use D channel signaling with SS7
- Will use SS7 call control with minimum call set-up time and efficient satellite BW utilization

### BSIS Broadband Service ISDN Satellite

- Advanced ISDN Satellite Design with onboard Class 5 Switch and SS7 Network Interface and layered protocol
- Provide Broadband ISDN Services (Primary Rate Access)
- Provide nationwide single hop, multiple high gain hopping beams, forward error control, optical processing, and "zero delay" satellite link interexchange node connectivity
- Will use D channel signaling with SS7
- Will center design around ATM fast packet switching techniques

Figure 1.1-1 NASA/SCAR Approaches for Advanced ISDN Satellites

ACTS will be controlled by a Master Ground Station (MGS) shown in Figure 1.2-1, "Closed User-Oriented Scenario". A user of the ACTS satellite orbiting switch request services from the MGS, a combination of the NASA Ground Station (NGS) and the Master

Control Station (MCS). The MGS, in turn, commands the satellite to switch the appropriate communication channel.

The ultimate aim of this element of the SCAR Program is to move these MGS functions on-board the next generation ISDN communications satellite as shown in Figure 1.2-2, "Advanced ISDN Satellite" as part of the FSIS architecture. The technical and operational parameters for the advanced ISDN communications satellite design will be obtained from the simulation of ISIS and FSIS engineering software models of the major subsystems of the ISDN communications satellite architecture. Discrete event simulation experiments will be performed with these model using various traffic scenarios, design parameters, and operational procedures. The data from these simulations will be analyzed using the NASA SCAR performance measures discussed in previous reports.

### **1.3 Document Overview**

This task completion report begins by citing the SCAR Simulation Objectives in Section 2, and explain the logic for the four simulation phases. A description of each phase is provided. Section 3 discusses the SCAR simulation development and methodology used to determine the design parameters for the SCAR advanced ISDN communications satellite design. Particular attention is given to the comparison between the ISIS and FSIS simulations since over 80% of the software modules will be reused between them. The two main sections of this task completion report are Section 4.3, Simulation Structures, which include a descriptions of both ISIS and FSIS architectures down to the process level and Section 4.4, Simulation Processes, which provides a detail description of each process.

Several appendices are included to provide more detail on the Scenario Traffic File (STF), Process Array Structure, the Traffic Model Database, the Q.931 Protocol Simulation, and the Measurement Save (MSave) file.



Government  
Systems

NASA - SCAR

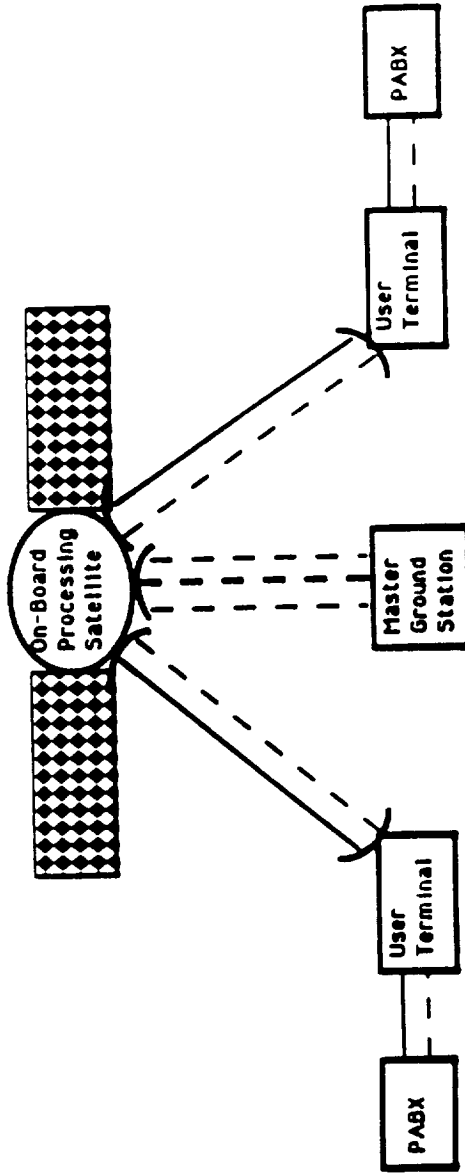


Figure 1.2-1 Closed User-Oriented Scenario

Ref Fournon 13155

Control Station (MCS). The MGS, in turn, commands the satellite to switch the appropriate communication channel.

The ultimate aim of this element of the SCAR Program is to move these MGS functions on-board the next generation ISDN communications satellite as shown in Figure 1.2-2, "Advanced ISDN Satellite" as part of the FSIS architecture. The technical and operational parameters for the advanced ISDN communications satellite design will be obtained from the simulation of ISIS and FSIS engineering software models of the major subsystems of the ISDN communications satellite architecture. Discrete event simulation experiments will be performed with these model using various traffic scenarios, design parameters, and operational procedures. The data from these simulations will be analyzed using the NASA SCAR performance measures discussed in previous reports.

### **1.3 Document Overview**

This task completion report begins by citing the SCAR Simulation Objectives in Section 2, and explain the logic for the four simulation phases. A description of each phase is provided. Section 3 discusses the SCAR simulation development and methodology used to determine the design parameters for the SCAR advanced ISDN communications satellite design. Particular attention is given to the comparison between the ISIS and FSIS simulations since over 80% of the software modules will be reused between them ..The two main sections of this task completion report are Section 4.3, Simulation Structures, which include a descriptions of both ISIS and FSIS architectures down to the process level and Section 4.4, Simulation Processes, which provides a detail description of each process.

Several appendices are included to provide more detail on the Scenario Traffic File (STF), Process Array Structure, the Traffic Model Database, the Q.931 Protocol Simulation, and the Measurement Save (MSave) file.



Government  
Systems

NASA - SCAR

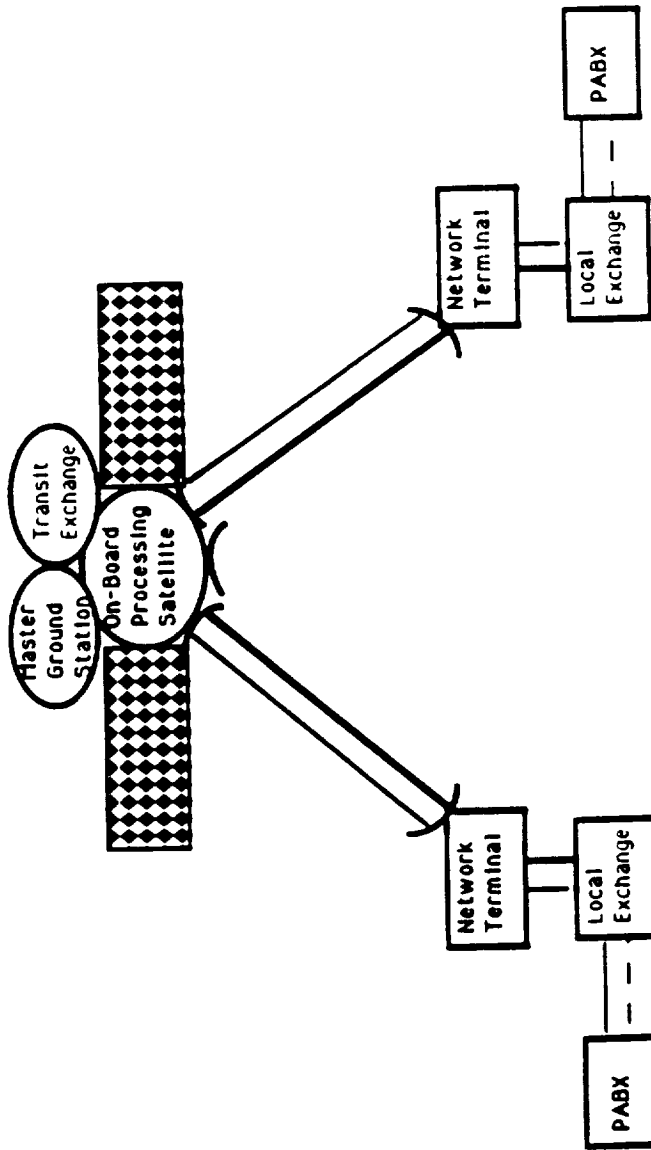


Figure 1.2-2 Advanced ISDN Satellite

## SECTION 2

### SCAR SIMULATION

#### 2.1 SCAR Simulation Objective

The objective of this SCAR simulation project is to design and develop software models that can be used to simulate those aspects of an ISDN communications satellite with sufficient fidelity to assist in determining design parameters. This simulation effort will assist in the development new advanced on-board satellite capabilities that will enable the provision of new services of an interim and full ISDN communication satellite. Figure 2.1-1., "ISDN Communications Satellite Simulation Top View" indicate the inputs and outputs expected of the SCAR simulation as well as the characteristics of the simulation, itself.

ISDN protocols, procedures, standards and user traffic form the input basis for the simulation. Together with the SS7 technology, the OSI methodology, and the satellite environment this ISDN aspect of communication satellite design is challengingly constrained. The SCAR ISDN communication satellite simulation uses discrete event based simulation of communication protocol flows through an engineering model to generated results traceable to the technical design parameters. The SCAR simulation outputs will be capable of demonstrating the viability of an ISDN satellite design and provide the rationale for recommending specific engineering parameters and changes to published ISDN standards.

#### 2.2 Simulation Programs

This end-to-end simulation is divided into four distinct simulation phases: database generation, scenario generation, simulation run, and product generation as shown in Figure 2.2-1., "SCAR Simulation Phases". The ISDN communications satellite end-to-end simulation is shown in Figure 2.2-2, "End to End Model Architecture". Each program is physically and functionally separated by input/output data files. This separation ensures that each simulator program is independent. The only link between these programs is the data file they share. Each program is briefly described in the following sections in order to provide an overview of the simulation process.

##### 2.2.1 Database Generation Program

The Database Generation (DbGen) program assembles the major ISDN user characteristics into a machine readable database. For this NASA SCAR effort the traffic model consists of a number of databases: the City Reference DB, ISDN User vs Industry DB, Application vs Industry DB, Application vs Time DB, and Application vs Bearer Services DB. Figure 2.2.1-1, "CONUS City Locations for NASA SCAR Traffic Model Database", shows the cities that are part of the traffic model. Those cities outlined with an ellipse identify the ACTS-east cities. Those cities outline with a rectangle identify the "ACTS-west" cities and the blackened squares depict the fixed antenna cities. The east/west city clusters are separated by a dashed line. The figure shows that the NASA SCAR traffic model is well aligned with the cities of interest for ACTS. The traffic model database represents the ISDN traffic for these cities and is the principal input to the scenario generation process. The traffic model databases data are presented in Appendix C, "SCAR Traffic Model Database Data"





Government Systems

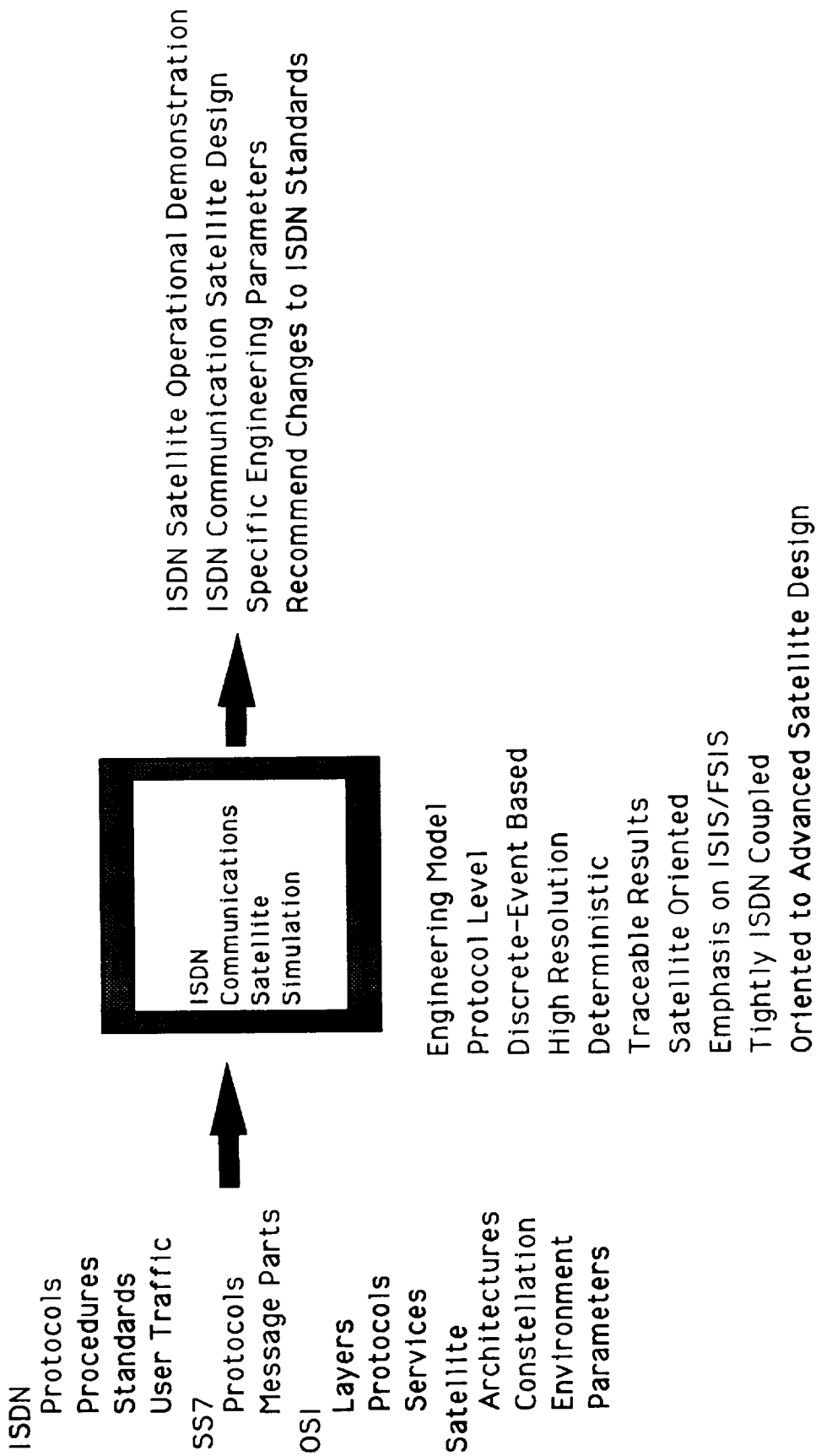


Figure 2.1-1 ISDN Communications Satellite Simulation Top View



Government Systems

NASA - SCAR

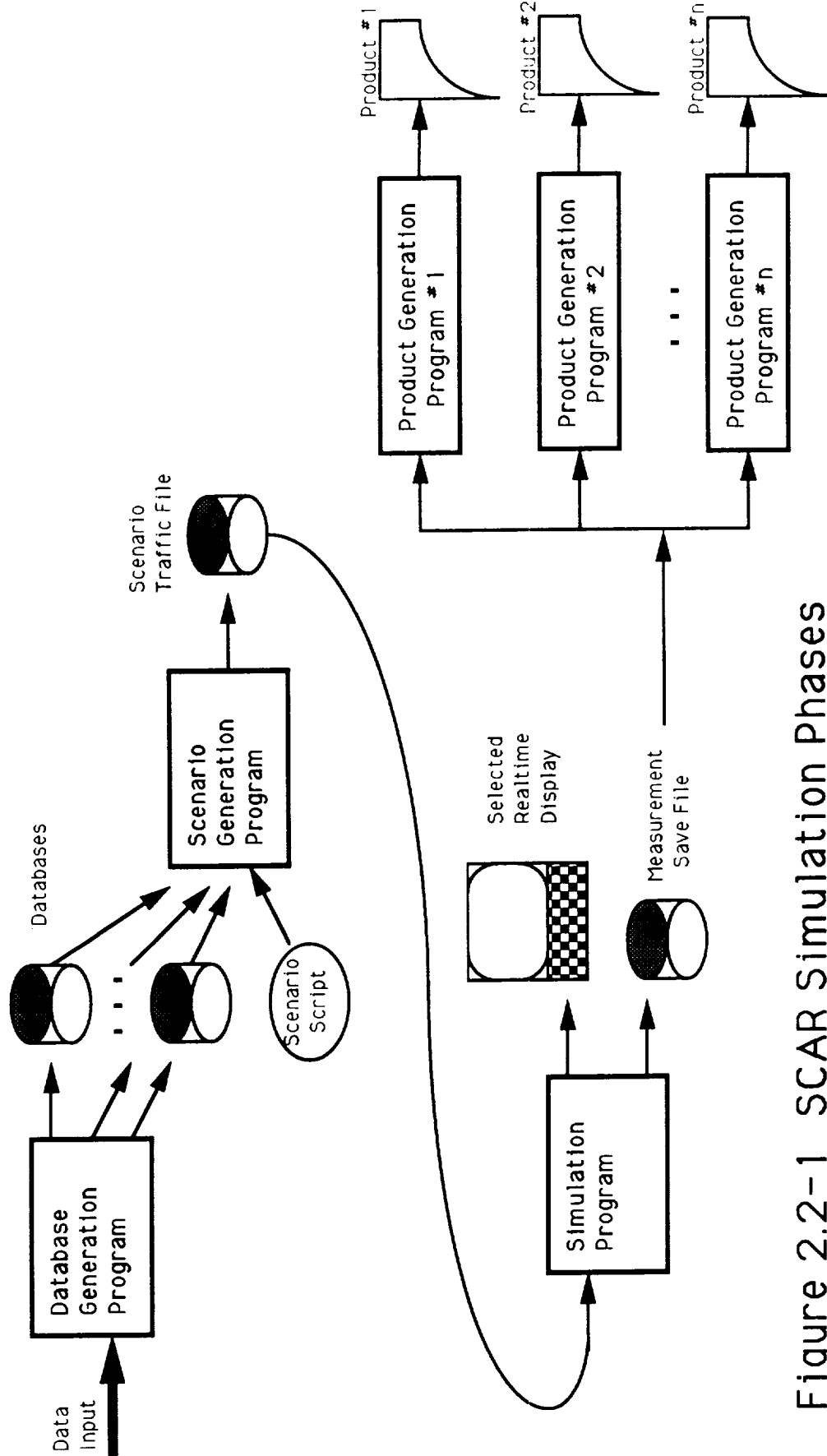


Figure 2.2-1 SCAR Simulation Phases



Government Systems

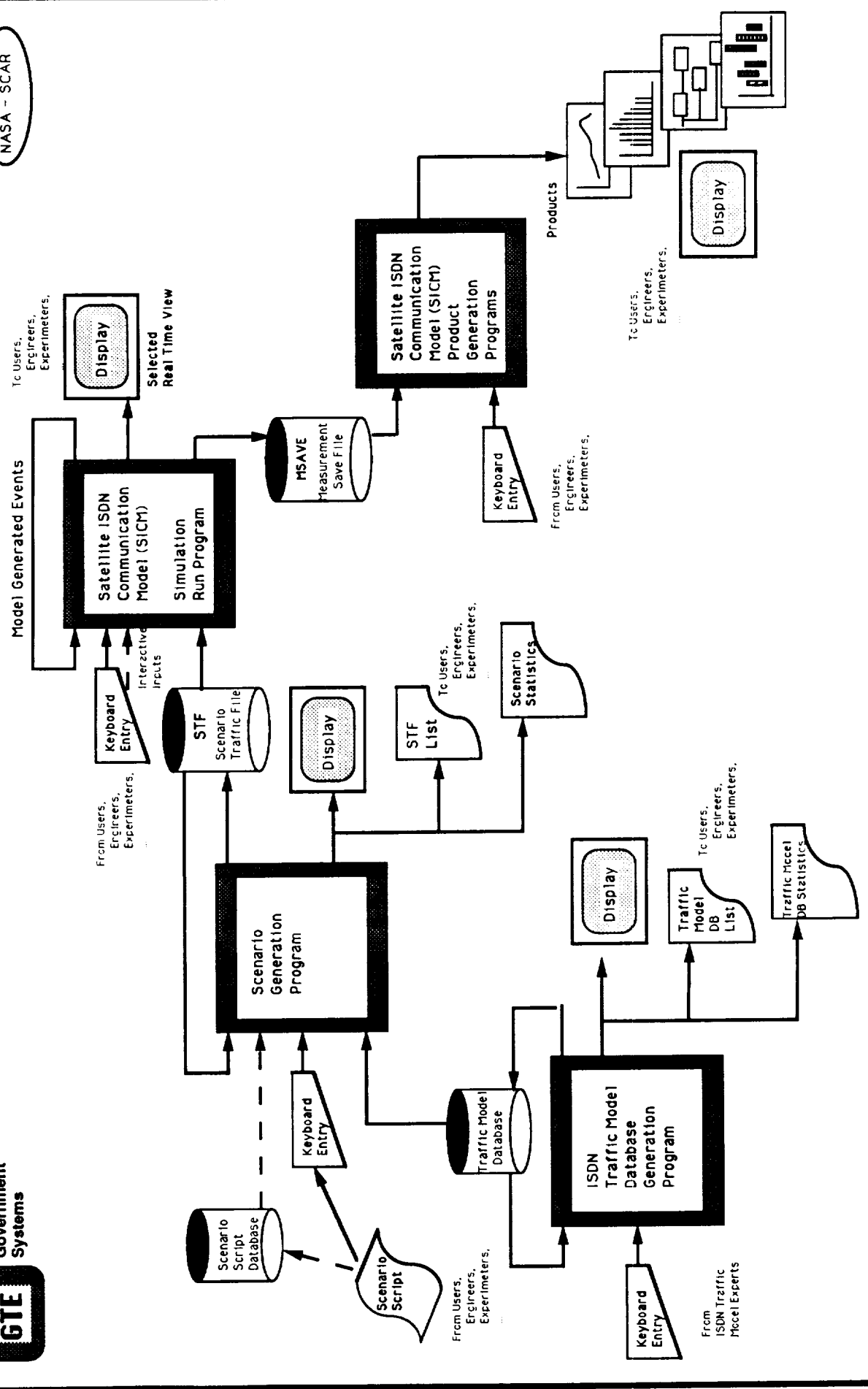


Figure 2.2-2 End to End ISDN Model Architecture



Government Systems

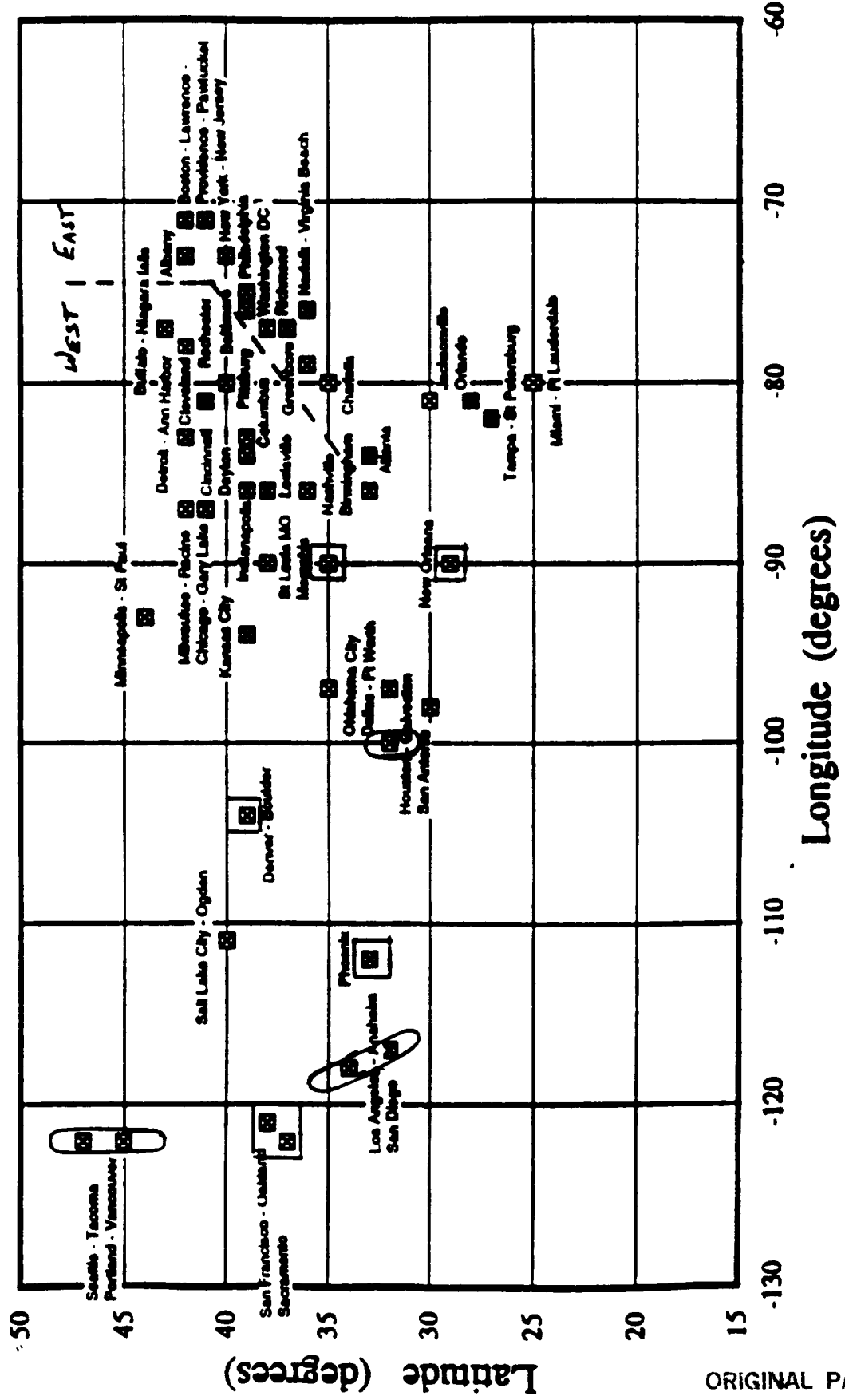


Figure 2.2.1-1 CONUS City Locations for NASA SCAR Traffic Model Database

5 T036 SCA DAT  
TraModel City Loc Map  
September 26, 1991

ORIGINAL PAGE IS OF POOR QUALITY

### **2.2.2 Scenario Generation Program**

The Scenario Generation (ScenGen) program selects the traffic model database entries that describes a scenario of ISDN users together with the statistical information of the ISDN services requested. The ScenGen program uses entries from the user traffic model database and engineering parameter databases to generate a list of time ordered, initiating discrete events. The discrete event list is call a Scenario Traffic File (STF). The STF is used to initialize the model for a specific advanced ISDN communications satellite design and to exercise that satellite design using the requests for ISDN services dictated by the ISDN user traffic.

### **2.2.3 Simulation Run Program**

The Simulation Run (SimRun) program consists of a model of the real world communications network of the major ISDN communications satellite components. Each of these ISDN communication components is represented by a block diagram within the overall architecture. As shown in Figure 2.2.3-1, "Simulation Run Program", the SimRun program essentially reads each discrete event from the (STF), takes the appropriate action, and logs that action and the corresponding results in a measurement save (MSave) file. The appropriate action taken by the simulation includes allocating and releasing communication resources, denying specific services, and calling other processes in-turn.

### **2.2.4 Product Generation Program**

The Product Generation (ProdGen) program reads the data in the MSave file and analyzes these data in accordance with specific algorithms. It is envisioned that there will be as many product generation programs as there are ISDN communications satellite issues to be studied: throughput, response time, trace, delay, call blocking, busy-minute, busy-hour, etc. Each ProdGen program is tailored to a particular area of ISDN communications satellite design. Performance measures will be used as criteria to evaluate the design parameters, operational procedures and degree of ISDN communications standard compliance of the particular ISDN communications satellite design.

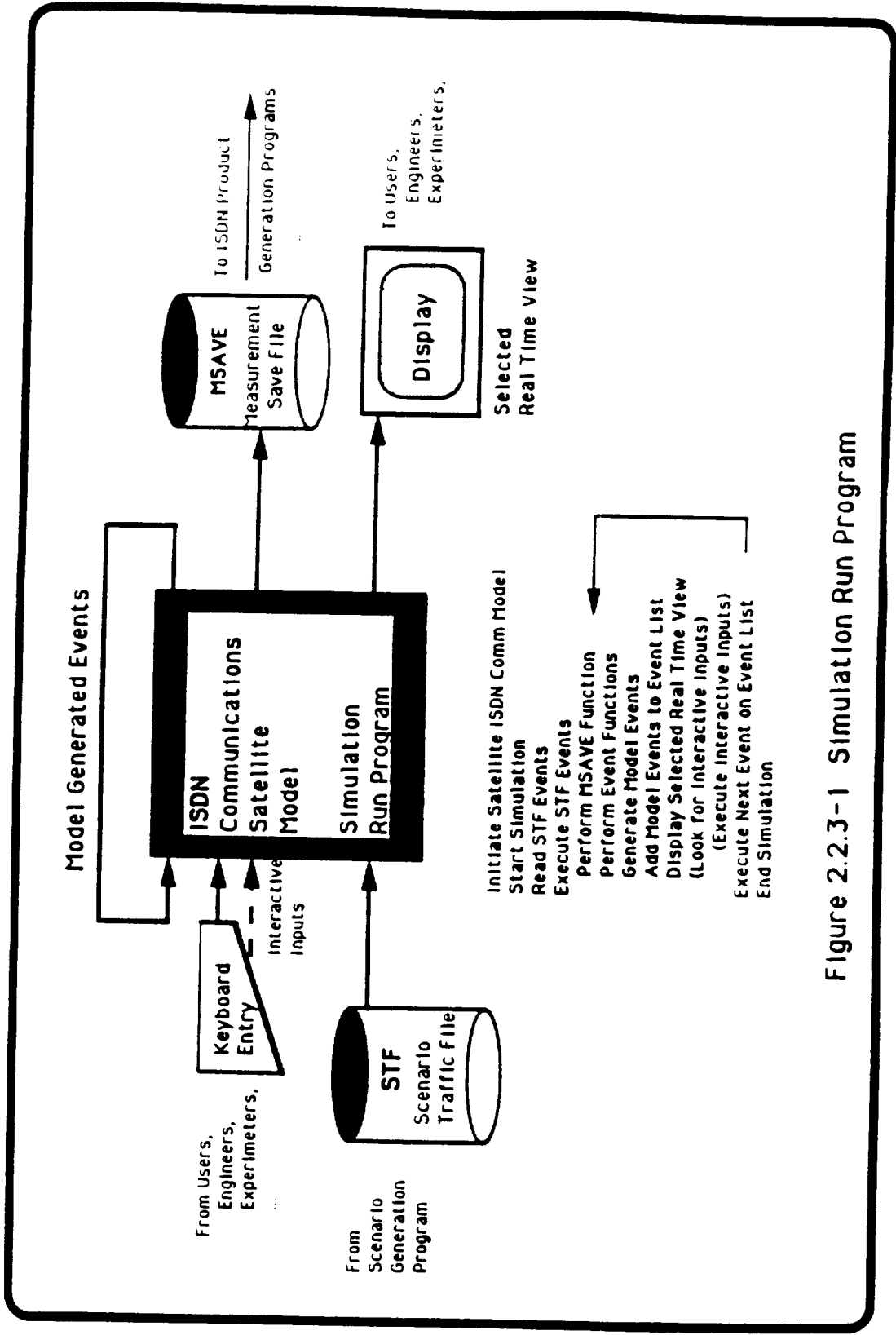


Figure 2.2.3-1 Simulation Run Program

## SECTION 3

### SIMULATION DEVELOPMENT

#### 3.1 Technical Concept

This NASA SCAR effort uses network modeling and simulation as the principal vehicle for determining the parameters of an ISDN communications satellite design. This network modeling and simulation must clearly delineate the network architecture (as defined by the network methodology, the number and kinds of communications elements and how those elements are configured), the network operations (including link and network layer protocols and how network functions are distributed among communications elements), and the system constraints (imposed both by the network and by the operational requirements).

#### 3.2 Network Model and Simulator Design

Discrete event simulator designs for both ISIS and FSIS were initially defined based on the Phase I network model of these communications architectures. The traffic model database and the scenarios, also developed in Phase I, were used to define ISIS and FSIS the designs using these simulator inputs. The simulator design outputs were based on the performance measures established in Phase I and will be used to evaluate overall ISDN communications satellite system design.

#### 3.3 Simulator Development

ISDN is based on the Open System Interconnection (OSI) Reference Model. Though the simulator design will focus on the second (data link) and third (network) layers to evaluate the performance of routing, acknowledgement, congestion control, and other protocol driven functions, this design will also address the time-out issues relate to the first (physical) layer. Although physical characteristics of the system will not be directly simulated, the effects of the physical conditions will be parametrically simulated. For example, the cases involving heavy rain, severe signal degradation, and higher error rates are examined in terms of protocol performance when multiple transmission are required. Instead of calculating a link budget where all signal losses and gains are summed and converting the signal-to-noise ratio and to a bit-error rate, the simulator will take the power loss associated for that error rate as input. Such simulation techniques reduce the complexity of the simulator design, while providing the same level of information about layer protocols and their timers.

#### 3.4 Generic Network Model for Simulation

Figure 3.4-1, "Generic Network Model Block Diagram", shows the major subsystems of a communications architecture for a generic ISDN communications satellite simulating two satellite terminals each supporting three users. For this simulation, these subsystem models associated with the satellite terminals consist of an uplink transmitter and transmitting antenna, a downlink receiver and receive antenna, three users generating traffic, and a multiplexer/demultiplexer that combines and separates this traffic. The satellite is modeled by corresponding receivers, transmitters, antennas, an on-board switch, and an on-board processor that decodes received commands, controls the switch, and generates response traffic.

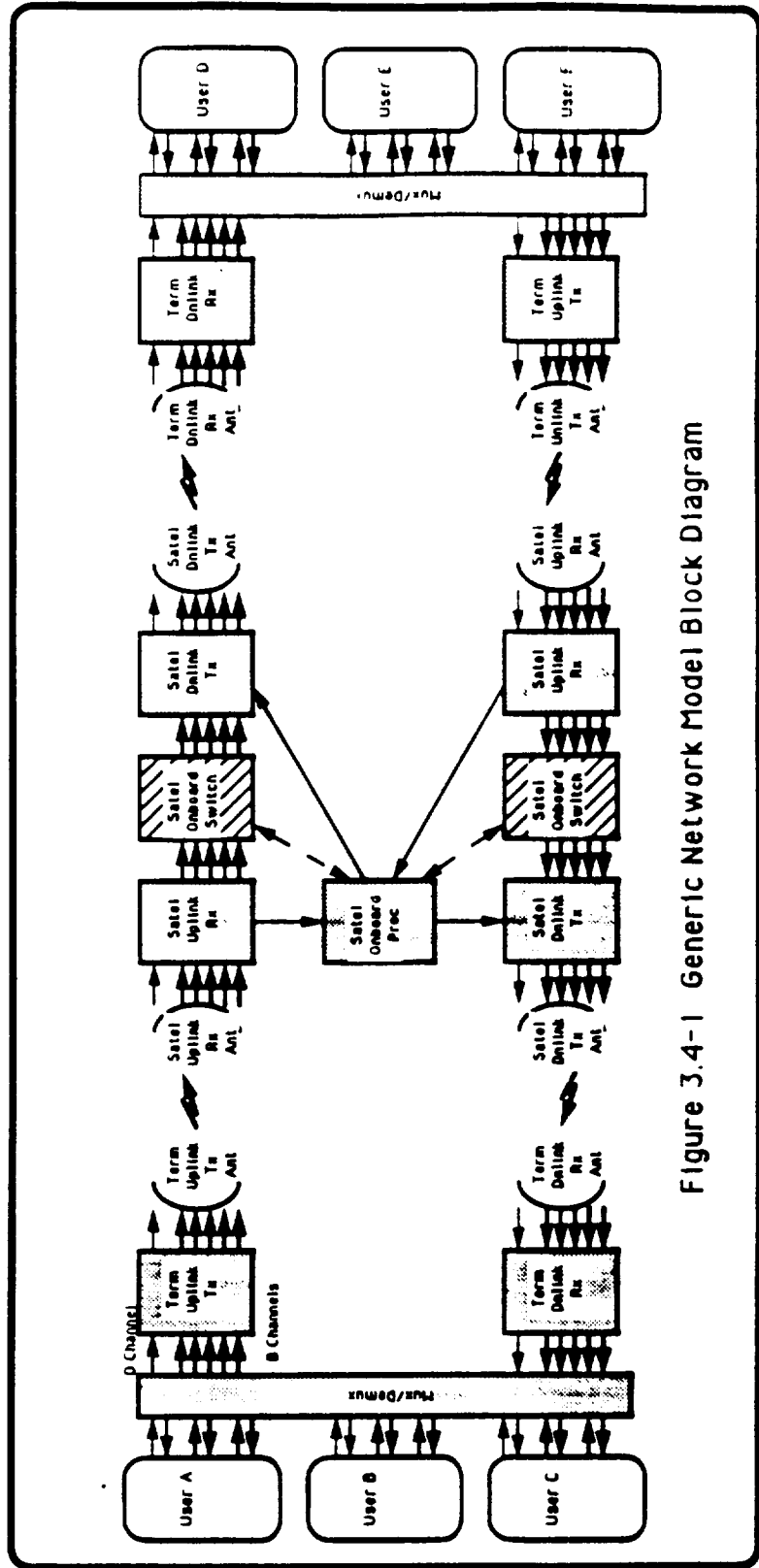


Figure 3.4-1 Generic Network Model Block Diagram



### 3.5 Generic Network Model Subsystems

Each of the communications subsystems in the network model design is represented by a software module that performs the functions of that communication component. Figure 3.5-1, "SCAR Network Model Systems", shows the generic network model presented in Figure 3.4-1 as interconnected software modules. Each module has parameters (p) inputs that determines that module's characteristics. For an antenna: p includes such things as the gain, beamwidth, scan rate, dwell time, etc. For a receiver: p includes frequency, burst rates, receiver threshold, receiver delay, etc. For a processor: protocol repertoire, processing time, clock frequency, number of ISDN resources, etc. In general each model module has a p-set that determines the design characteristics. These p's are input via an engineering data matrix that is loaded before the simulation run begins. These engineering data determine the design of the ISDN communication satellite subsystems being evaluated.

The initial discrete events (E) are part of the STF and are executed as a function of time depending on the scenario that generated them. Each of event (E) serves as the initial event for a sequence of internal protocols generated by a particular module in the simulation. Each module processes each discrete event (E) and takes actions accordingly. Many of these actions include generating response events (e) for another modules. The response events (e) are integrated in time order with the initial events (E) via the (stf) to be executed at their respective times. For ISDN protocols, a single initial discrete event (E) will generate many sequential response events (e). The simulation process continues the execution of the time ordered event list of Es and es until the simulation ends. The technical data generated by the simulation is obtained from a Measurement Save (MSave) file. Every time an discrete event is presented to a module its identity and its time of arrival is time-tagged and saved on the MSave file (M). Also, all resource allocations, resource releases, resource denials, event generations, and the status of every module are saved on the MSave file (M) together with their time of occurrence. The MSave file has a complete time ordered history of every event, action, and status of every module for the entire simulation. That MSave file can be analyzed on post run basis to generate any number of technical and operational report products.

### 3.6 Generic Network Model Simulation Software

The simulation software inside each module determines its communication characteristics and responses. Figure 3.6-1, "SCAR Network Model Simulation Software", depicts the software flow chart for a single module - Proc. In that example, when the processor function (Proc) receives the event (Sig#27). It first reports the event and time to MSave (M). The software then determines if the requested resources are available. At the beginning of the simulation parameter (P1) allocated a number of these resources to Proc. If none of those resources are now available, Proc sends a "No Resources Available" message to the MSave. Proc then clears all Sig#27 items and returns control to the simulation timing routine. On the other hand if resources were available, Proc would allocate and adjust those resources; report the allocation to MSave; and activate the next process in the sequence. The activation time for the next process will be calculated using the processing delay value of P2 milliseconds. Both P1 and P2 values were assigned via the engineering data matrix before the start of the simulation.

The same software modules will be re-used with different parametric values for similar functions such as antenna, receiver, processor, etc. for both the ISIS and FSIS architectures. This will result in the generation of less simulation code for a given software development.

NASA - SCAR

Government Systems

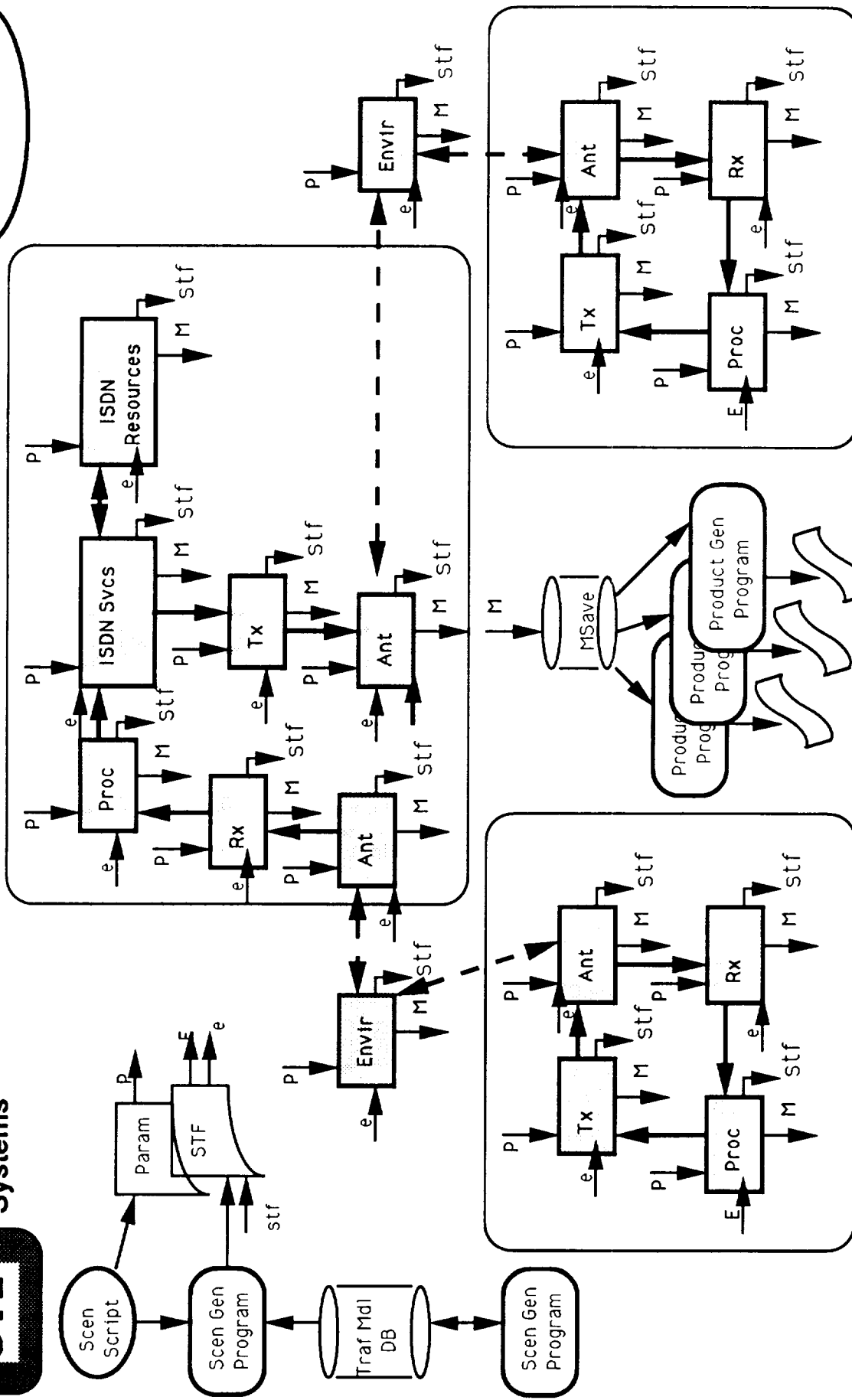


Figure 3.5-1 SCAR Network Model Systems

5 T036 SCAS DAT  
 SCAR Network Model Systems  
 September 13, 1991

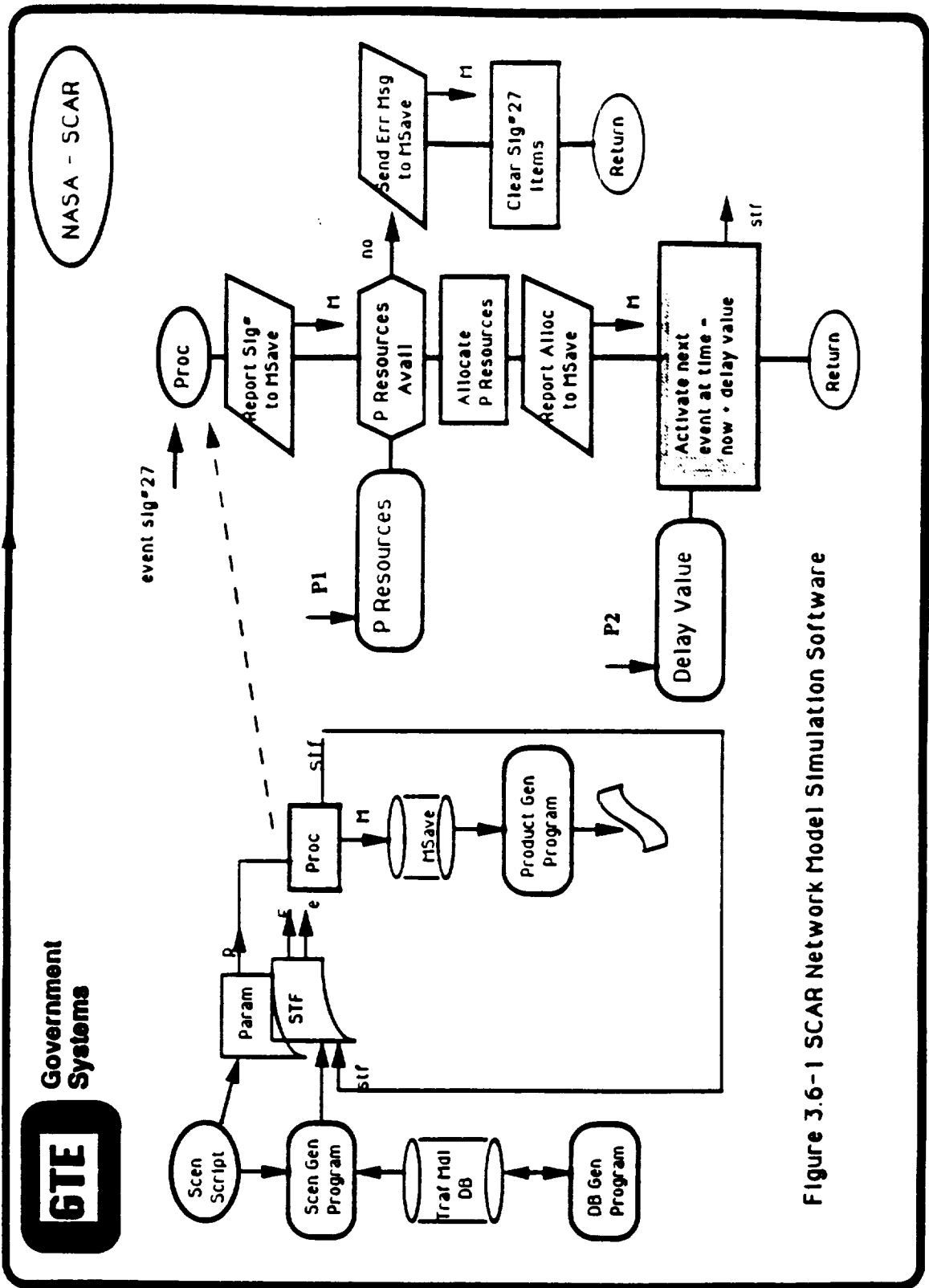


Figure 3.6-1 SCAR Network Model Simulation Software

### **3.7 Multi-Terminal SCAR Simulations**

The Figure 3.7-1, "FSIS Multi-Terminal SCAR Model", depicts a satellite-based switch using on-board control to simulate communications services between terminals on the left. This same model is also capable of simulating central offices on the right. Such simulations can be used as a vehicle for analyzing the protocol messages flow among all the users connected to the satellite.



Government Systems

NASA - SCAR

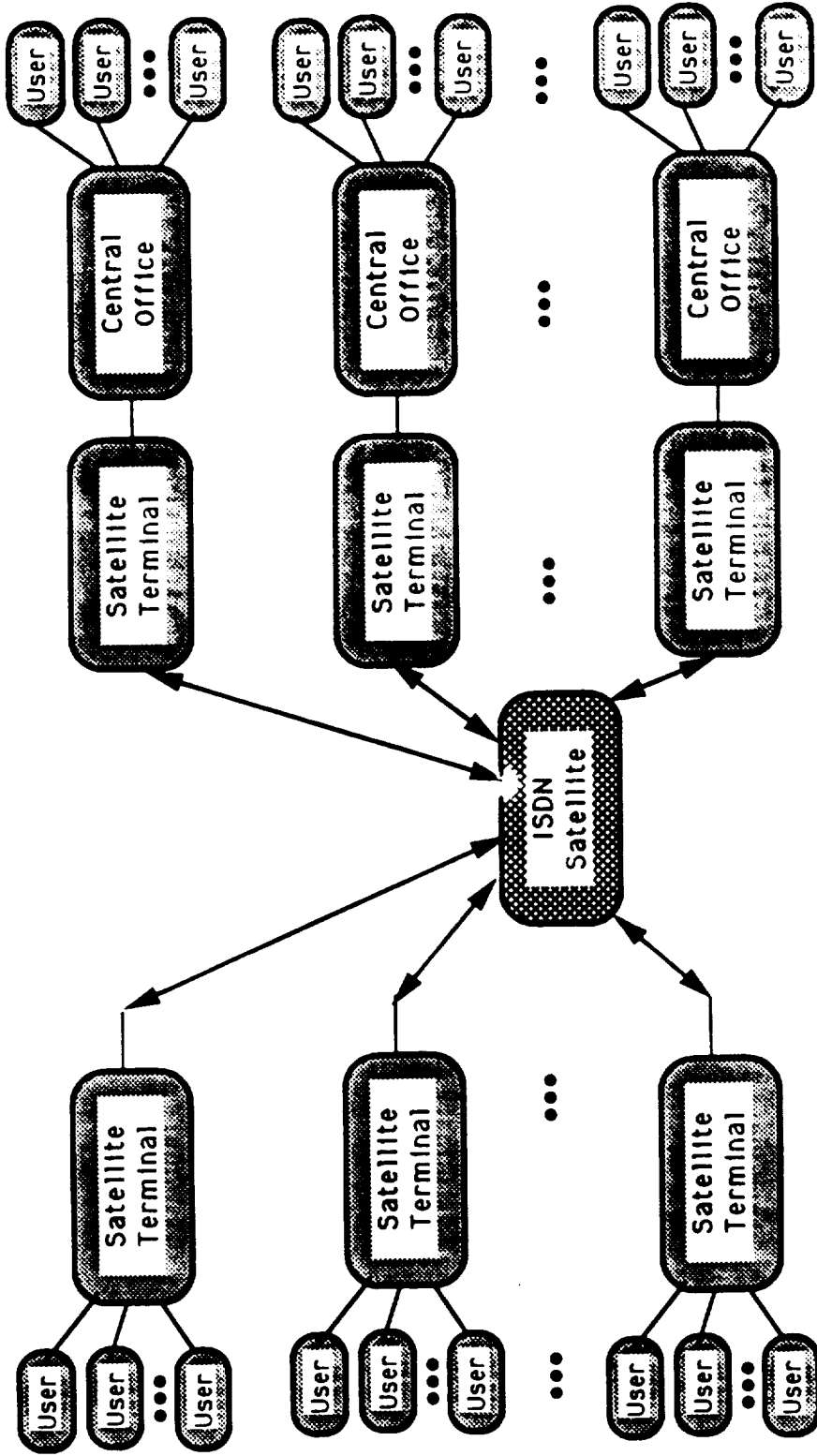


Figure 3.7-1 FSIS Multi-Terminal SCAR Model

## SECTION 4

### Simulation Components and Processes

#### 4.1 Introduction

The simulation context of the previous sections is now applied to the ISIS and FSIS architectures. The FSIS simulation focuses on the on-board satellite ISDN circuit switched protocols: call control (Q.931), LAPD (Q.921), PRI (I.431), BRI (I.430), and SS7 (ISUP). Whereas the ISIS simulation addresses ground based call management using these same protocols supplemented by special order-wire (OW) commands between the satellite and Master Control Station (MGS). The D-channel protocol messages and their associated timing, propagation, processing, and execution are the main concerns of both these models. The B-channels are modeled as resources to be allocated and released as their availability dictate.

As illustrated in Figure 4.1-1, "FSIS SCAR Model Systems", the FSIS system provides the ISDN user access via VSATs connected with ISDN Satellite Terminal Adapters (ISTAs). The FSIS simulation use D channel signalling and those parts of the ISUP necessary for call control. This approach enables an advanced satellite to provide nationwide ISDN using an on-board call control and B-channel switching architecture. The ultimate aim of this aspect of this SCAR Program is to move all ISDN functions on-board the satellite for the next generation ISDN communications satellite design.

In both cases, ISIS and FSIS, the simulation analyses will be obtained from engineering software models of their major subsystems of the ISDN communications satellite architecture and their appropriate ground terminations. Discrete event simulation experiments will be performed with these models using various traffic scenarios, design parameters, and operational procedures and performance measures.

#### 4.2 Definition and Purpose

Both ISIS and FSIS simulations consist of a number of VSATs connected to an ISDN satellite via a single hop. The VSATs will exchange ISDN traffic on a demand access, circuit switched basis. The purpose is to investigate the throughput, response time, blocking probability, and robustness of these two ISDN satellite architectures in a benign environment to provide a performance measures baseline and to investigate protocol timing issues at the lower layer levels. Particular attention will focus on the timing and time-outs associated with the ISDN physical layer protocol. These simulations will also deal with issues such as: packet switching on the B and D channels, full SS7 protocols, weather, and direct connectivity to an ISDN public switched network (IPSN).

#### 4.3 Simulation Structures

Both the ISIS and FSIS simulation will be described in the same context. A top view of the architecture is presented at the communication component level. This provides visibility into the architecture and links for these major communication components of the engineering models that are used to represent them. The next view treats these models as simulation processes and connects them in an end-to-end diagram representing the protocol flow. To ease the routing algorithm for the simulation a sequential number was ascribed to each process, Process Index (PI). This PI integer uniquely defines the specific occurrence of the process, its neighbors at that time, and the direction of protocol flow. The last view



Government Systems

NASA - SCAR

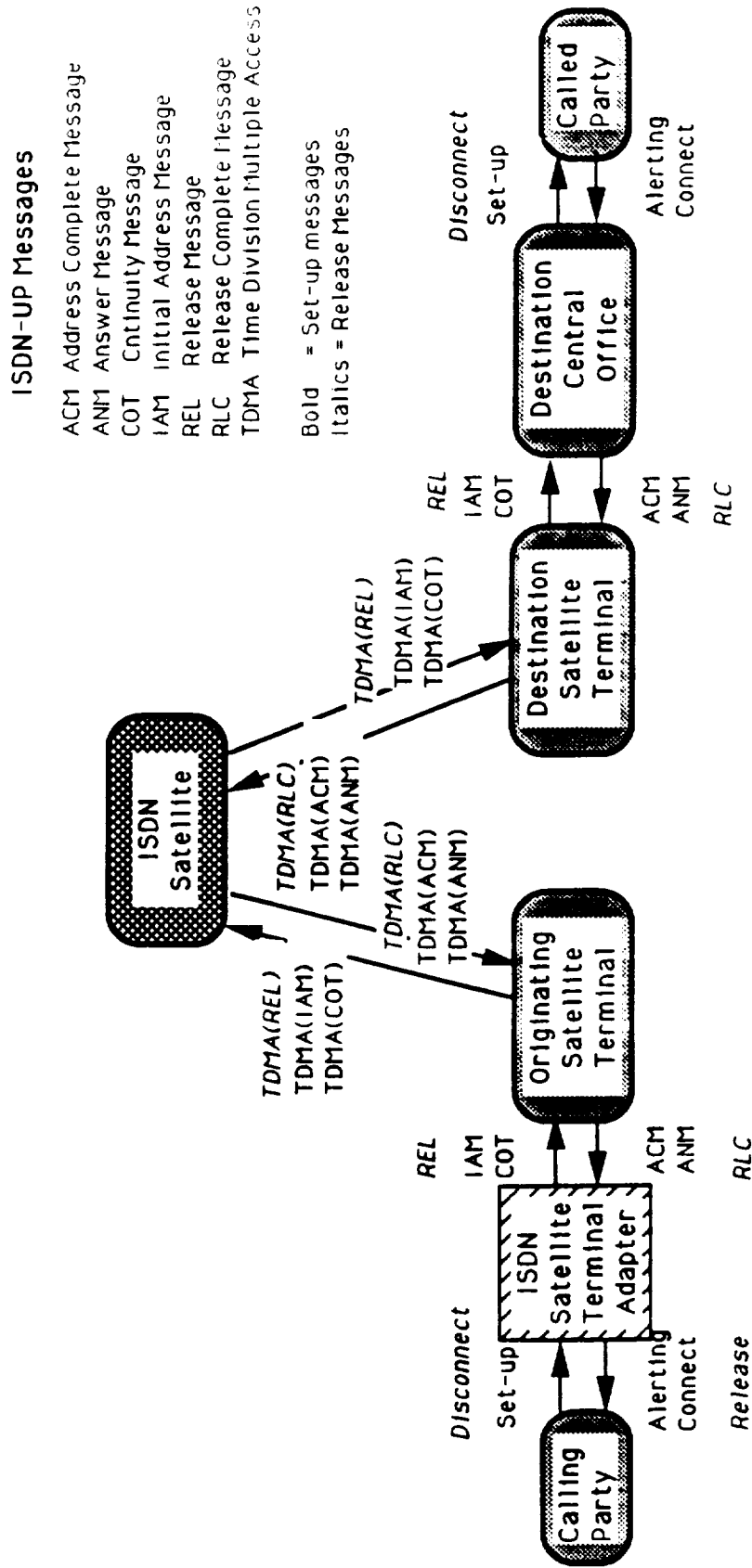


Figure 4.1-1 FSIS SCAR Model Systems

tabulates these processes into a pictorial matrix that associates each of them with their unique process index. The sequential aspects of this representation form a sort of process index "racetrack" pattern that can be used to visualize the protocol hand-off from one PI element to next.

**For the FSIS architecture**, seven major communication components are connected by six interfaces. Figure 4.3-1., "FSIS Simulation Communication Components", shows the ISDN Telephone, ISDN Satellite Terminal Adapter, VSAT Satellite Terminal and the FSIS Satellite connected by the S-Interfaces, U-Interfaces, and Propagation. Figure 4.3-2., "FSIS End-to-End Simulation Processes", are connected into a network using the Process Index (PI) as a sequence identification mechanism for tracking protocol flow. The processes are aligned with the major communication components depicted at the top of the page. Figure 4.3-3., "FSIS Simulation Communication Components and Model Processes Racetrack", lists all the simulation processes along with their PI numbers. The reference numbers are keyed to the text in this section that provide more detail about both the communication components and the processes. The FSIS simulation architecture statistics include:

- 4 types of major Communication Components
- 16 types of simulation modules (processes)
- 77 process indexes
- 4.8 factor of software reuse (77 / 16)

**For the ISIS architecture**, nine major communication components are connected by eight interfaces. Figure 4.3-4., "ISIS Simulation Communication Components", shows the ISDN Telephone, ISDN Satellite Terminal Adapter, VSAT Satellite Terminal, the FSIS Satellite, and the Master Ground Station connected by the S-Interfaces, U-Interfaces, and Propagation. Figure 4.3-5., "ISIS End-to-End Simulation Processes", are connected into a network using the Process Index (PI) as a sequence identification mechanism for tracking protocol flow. Figure 4.3-6., "ISIS Simulation Communication Components and Model Processes Racetrack", lists all the simulation processes along with their PI numbers. The reference numbers are keyed to the text in this section that provide more detail about both the communication components and the processes. The ISIS simulation architecture statistics include:

- 5 types of major Communication Components
- 18 types of simulation modules (processes)
- 109 process indexes
- 6.0 factor of software reuse (109 / 18)

The commonality factor between the ISIS and FSIS architectures is 89% (16 common modules of 18 modules). The following sections describes each of these ISIS and FSIS communication components in terms of their implementing modeling processes.

#### **4.3.1 ISIS and FSIS Satellite Communication Component**

The advanced ISDN communications satellite design under the NASA SCAR Program uses as its design starting point an ISDN switch in orbit. A user of the ISDN satellite requests services using ISDN protocols. These ISDN protocols are routed to the satellite via the VSATs and ISTAs . Depending on the communication satellite design, ISIS or FSIS, these ISDN protocols processed differently.



**GTE**

**Government  
Systems**

**NASA - SCAR**

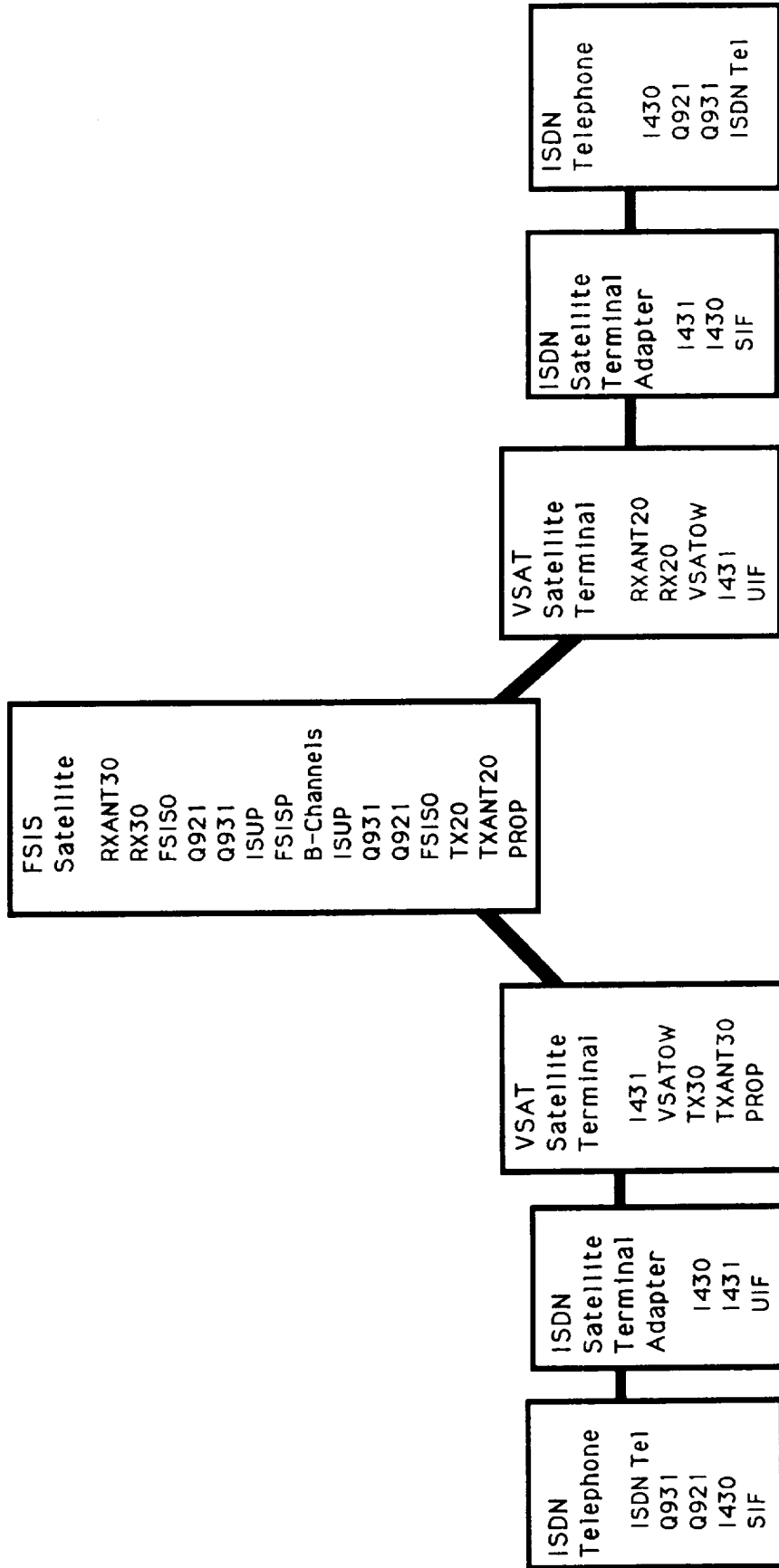


Figure 4.3-1 FSIS Simulation Communication Components



Comm Component:	Process	Model	Model	
Para # 4.3.5	Index (PI)	Processes:	FSIS Rpt	Processes:
ISDN Tel	1	ISDNO (Tel)	Para # 4.4.5	ISDNO (Tel) 77
	2	Q.931U (User)	4.4.11	Q.931U (User) ^ 76
	3	v Q.921U (User)	4.4.10	Q.921U (User)   75
	4	I430U	4.4.3	I430U   74
4.3.7	5	SIF	4.4.14	SIF 73
4.3.4				
ISTA	6	I430N	4.4.3	I430N 72
	7	I431U	4.4.4	I431U 71
4.3.8	8	UIF	4.4.17	UIF 70
4.3.3				
VSAT	9	I431N	4.4.4	I431N 69
	10	VSATO (OW)	4.4.18	VSATO (OW) ^ 68
	11	v Tx (30)	4.4.12	Rx (20)   67
	12	TxAnt (30)	4.4.13	RxAnt (20)   66
4.3.6	13	Propa (30)	4.4.9	Propa (20) 65
4.3.1				
<i>FSIS ISDN</i>	14	RxAnt (30)	4.4.16	TxAnt (20) 64
<i>Comm Satellite</i>	15	Rx (30)	4.4.15	Tx (20) 63
<i>Poss Bypass</i>	16	FSISO (OW)	4.4.2	FSISO (OW) 62
	17	Q.921N (Network)	4.4.10	Q.921N (Network) 61
	18	Q.931N (Network)	4.4.11	Q.931N (Network) 60
	19	ISUP	4.4.6	ISUP ^ 59
<i>CCI/ResMgt/ResAlloc</i>	20	FSISP (Proc) BCh	4.4.1	FSISP (Proc)   58
	21	v ISUP	4.4.6	ISUP   57
	22	Q.931N (Network)	4.4.11	Q.931N (Network) 56
	23	Q.921N (Network)	4.4.10	Q.921N (Network) 55
<i>Poss Bypass</i>	24	FSISO (OW)	4.4.2	FSISO (OW) 54
	25	Tx (20)	4.4.12	Rx (30) 53
	26	TxAnt (20)	4.4.13	RxAnt (30) 52
4.3.6	27	Propa (20)	4.4.9	Propa (30) 51
4.3.3				
VSAT	28	RxAnt (20)	4.4.16	TxAnt (30) 50
	29	Rx (20)	4.4.15	Tx (30) ^ 49
	30	VSATOW	4.4.18	VSATOW   48
	31	v I431N	4.4.4	I431N   47
4.3.8	32	UIF	4.4.17	UIF 46
4.3.4				
ISTA	33	I431U	4.4.4	I431U 45
	34	I430N	4.4.3	I430N 44
4.3.7	35	SIF	4.4.14	SIF 43
4.3.5				
ISDN Tel	36	I430U	4.4.3	I430U 42
	37	Q.921U (User)	4.4.10	Q.921U (User) ^ 41
	38	Q.931U (User)	4.4.11	Q.931U (User)   40
	39	v ISDND (Tel)	4.4.5	ISDND (Tel)

Figure 4.3-3. FSIS Simulation Communication Components and Model Processes Racetrack



Government Systems

NASA - SCAR

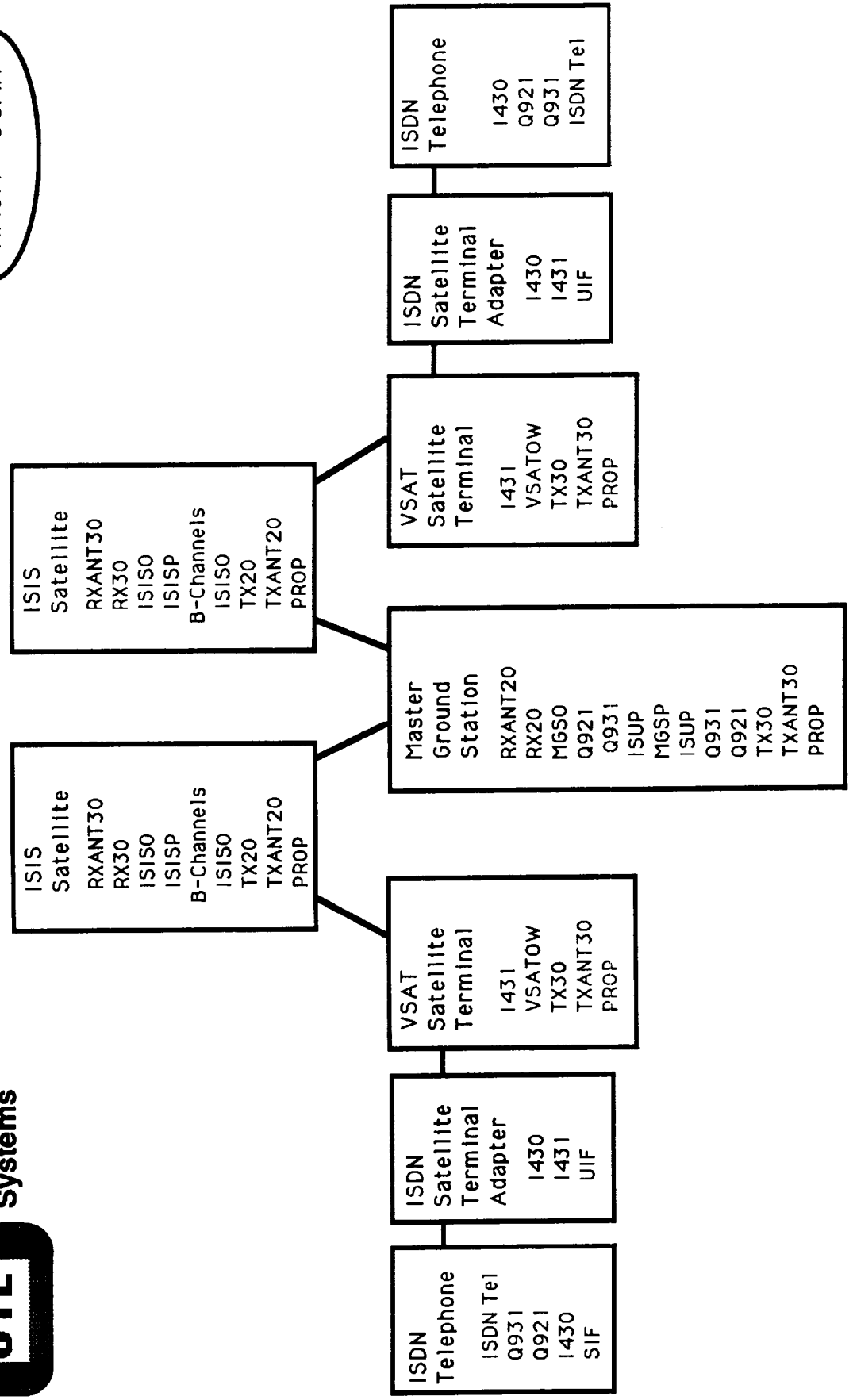


Figure 4.3-4 ISIS Simulation Communication Components



Comm Component:	Process	Model	Model		
Para # 4.3.5	Index (PT)	Processes:	Sim Rpt	Processes:	PT
ISDN Tel	1	ISDNO (Tel)	4.4.5	ISDNO (Tel)	109
	2	Q.931U (User)	4.4.11	Q.931U (User)	108
	3	Q.921U (User)	4.4.10	Q.921U (User)	107
	4	I430U	4.4.3	I430U	106
4.3.7	5	SIF	4.4.14	SIF	105
4.3.4	ISTA	6	I430N	4.4.3	I430N
		7	I431U	4.4.4	I431U
4.3.8	8	UIF	4.4.17	UIF	102
4.3.3	VSAT	9	I431N	4.4.4	I431N
		10	VSATO (OW)	4.4.18	VSATO (OW)
		11	Tx (30)	4.4.12	Rx (20)
		12	TxAnt (30)	4.4.13	RxAnt (20)
4.3.6	13	Propa (30)	4.4.9	Propa (20)	97
4.3.1	ISIS ISDN Comm Satellite	14	RxAnt (30)	4.4.16	TxAnt (20)
		15	Rx (30)	4.4.15	Tx (20)
		16	ISISO (OW)	4.4.2	ISISO (OW)
		17	ISISP (Proc)	4.4.1	ISISP (Proc)
		18	ISISO (OW)	4.4.2	ISISO (OW)
		19	Tx (20)	4.4.12	Rx (30)
		20	TxAnt (20)	4.4.13	RxAnt (30)
4.3.6	21	Propa (20)	4.4.9	Propa (30)	89
4.3.2	Master Ground Station MGS	22	RxAnt (20)	4.4.16	TxAnt (30)
		23	Rx (20)	4.4.15	Tx (30)
		24	MGSO (OW)	4.4.7	MGSO (OW)
		25	Q.921N (Network)	4.4.10	Q.921N (Network)
		26	Q.931N (Network)	4.4.11	Q.931N (Network)
		27	ISUP	4.4.6	ISUP
		28	MGS (Proc)	4.4.8	MGS (Proc)
		29	ISUP	4.4.6	ISUP
		30	Q.931N (Network)	4.4.11	Q.931N (Network)
		31	Q.921N (Network)	4.4.10	Q.921N (Network)
		32	MGSO (OW)	4.4.7	MGSO (OW)
		33	Tx (30)	4.4.12	Rx (20)
		34	TxAnt (30)	4.4.13	RxAnt (20)
		4.3.6	35	Propa (30)	4.4.9
4.3.1	ISIS ISDN Comm Satellite	36	RxAnt (30)	4.4.16	TxAnt (20)
		37	Rx (30)	4.4.15	Tx (20)
		38	ISISO (OW)	4.4.2	ISISO (OW)
		39	ISISP (Proc)	4.4.1	ISISP (Proc)
		40	ISISO (OW)	4.4.2	ISISO (OW)
		41	Tx (20)	4.4.12	Rx (30)
		42	TxAnt (20)	4.4.13	RxAnt (30)
4.3.6	43	Propa (20)	4.4.9	Propa (30)	67
4.3.3	VSAT	44	RxAnt (20)	4.4.16	TxAnt (30)
		45	Rx (20)	4.4.15	Tx (30)
		46	VSATOW	4.4.18	VSATOW
		47	I431N	4.4.4	I431N
4.3.8	48	UIF	4.4.17	UIF	62
4.3.4	ISTA	49	I431U	4.4.4	I431U
		50	I430N	4.4.3	I430N
4.3.7	51	SIF	4.4.14	SIF	59
4.3.5	ISDN Tel	52	I430U	4.4.3	I430U
		53	Q.921U (User)	4.4.10	Q.921U (User)
		54	Q.931U (User)	4.4.11	Q.931U (User)
		55	ISDND (Tel)	4.4.5	ISDND (Tel)

Figure 4.3-6 ISIS Simulation Communication Components and Model Processes Racetrack

**For ISIS**, the ISDN satellite operations are modeled by an uplink receiving antenna (RxAnt 30) and receiver (Rx 30) that are connected to the on-board ISDN orderwire processor (ISISO). See Figure 4.3-6. The ISISO either routes the protocol messages to the VSAT satellite downlink or to the MGS satellite downlink. The MGS acts on all protocol messages destined for this satellite. The ISIS on-board processing acts on the commands from the MGS to switch the allocated B-channels as directed by order wire commands. Both the VSAT and MGS downlinks are modeled by a downlink transmitter (Tx20) and an associated downlink transmitting antenna (TxAnt20). Both the downlink and uplink propagation are modeled by propagation (Prop) process that delays these protocol messages as a function of distance between the satellite and the respective ground terminal.

**For FSIS**, the ISDN satellite operations are modeled by an uplink receiving antenna (RxAnt 30) and receiver (Rx 30) that are connected to the on-board ISDN orderwire processor (FSISO). See Figure 4.3-1. The FSISO either routes the protocol messages to the ISDN satellite downlink or routes them through the protocol conversion modules: Q921N, Q931N, and ISUP to the on-board protocol processor (FSISP). The FSISP acts on all protocol messages destined for this satellite. Reply protocol messages follow a reverse protocol excursion back to their destination. The ISDN downlink is modeled by a downlink transmitter (Tx20) and an associated downlink transmitting antenna (TxAnt20). Both the downlink and uplink propagation are modeled by propagation (Prop) process that delays these protocol messages as a function of distance between the satellite and the ground terminal.

#### **4.3.2 Master Ground Station (MGS) Communication Component**

The ISDN Master Ground Station (MGS) serves as the ground based ISDN protocol analysis and decision component for satellite not capable of performing these functions. The MGS operations are modeled by a downlink receiving antenna (RxAnt 20) and receiver (Rx 20) that are connected to the ground based order wire processor (MGSO). See Figure 4.3-6. The MGSO either routes the protocol messages to the ISDN satellite uplink or routes them through the protocol conversion modules: Q921N, Q931N, and ISUP to the ground based protocol processor (MGSP). The MGSP acts on all protocol messages destined to it.. Reply protocol messages follow a reverse protocol excursion back to their destination. The MGS uplink is modeled by an uplink transmitter (Tx30) and an associated uplink transmitting antenna (TxAnt30). Both the downlink and uplink propagation are modeled by propagation (Prop) process that delays these protocol messages as a function of distance between the satellite and the MGS.

#### **4.3.3 VSAT Communication Component**

The VSAT user terminal represents the ISDN user entry into the ISDN communications satellite. For both the ISIS and FSIS architecture, the VSAT is generic terminal capable of converting I431 protocol to uplink signals and converting downlink signals to I431 protocol. The VSAT connects the user with U-interface and connects to the ISDN satellite with a propagation (Prop) interface. As such the VSAT represents the exchange termination (ET) for the user. The VSAT converts ISDN protocol messages into TDMA uplink signals in one direction and converts the downlink signals to ISDN protocols in the other direction.

The VSAT operations are modeled by a TDMA downlink receiving antenna (RxAnt20) and receiver (Rx20) that are connected to the VSAT orderwire processor (VSATO). The VSATO translates all downlink signals into I431 protocols messages. The I431 process provides the 1,544 kbps primary rate ISDN interface at the U-interface level.

The VSAT TDMA uplink operations are modeled in similar manner to convert ISDN protocols to uplink signals. The ISDN protocols come to the I431 process via the U interface. The VSATO converts the I431 protocol into a TDMA format for uplink transmission via Tx30 and TxAnt30 to ISDN communications satellite. Both the downlink and uplink propagation are modeled by the propagation (Prop) process that delays the protocol messages as a function of distance between the satellite and the VSAT.

#### **4.3.4 ISDN Satellite Terminal Adapter (ISTA) Communication Component**

The ISDN satellite terminal adapter (ISTA) represents the user's NT2 and NT1 connection between the user at the S-interface and the exchange termination (ET) at the U-interface. It represents protocol conversion necessary for aggregating a number of BRI services in a PRI link for ultimate translation into a TDMA uplink. For a downlink the ISTA also converts a PRI connection into a BRI service connections.

The ISTA operations are modeled by a Layer 1, physical protocol conversion process (I430) process at the S-interface. These protocols are converted up and down the OSI layers to match the S-interface BRI protocols to the U-interface PRI protocols. The translation process converts I430 protocols into I431 protocols in the S-interface to U-interface direction. The reverse sequence of processes models the U-interface to S-interface direction.

#### **4.3.5 ISDN Telephone User Communication Component**

For the FSIS Network Model the ISDN telephone represents the source and sink of all ISDN call connections. The off-hook and on-hook conditions are used as a starting point for the call connection protocol sequences that are converted along the OSI layer chain to the S-interface of the network termination (NT).

The ISDN Telephone operations are modeled by a human interface process (ISDNO and ISDND) that provides the on-hook and off-hook conditions. The ISDNO process act as originator of the Layer 3 protocol sequence using the Q931 messages that are converted down the OSI layers by the Q931, Q921 and I430 processes to S-Interface signals. That sequence is triggered by the initiating Request and Terminating events on the STF. The ISDND process represents the destination user in the same way as ISDNO portrays the originator. The reverse sequence of protocol processes models the S-interface to ISDND direction.

#### **4.3.6 Propagation Communication Component**

Both the downlink and uplink propagation in both ISIS and FSIS architectures account for the time delay experienced by a signal as it propagates between the ISDN satellite and any ground terminal. A significant amount of time is spent in signal propagation. Propagation is modeled by a single propagation (Prop) process that delays the signal as a function of distance between the satellite and the ground terminal. That distance depends on the satellite orbit and topology and the terminal distribution. These propagation distances change dramatically as a function of time and points of origin and destination. For a particular simulation these satellite to earth station distances may pre-calculated and Stored as part of the City array. Similarly, transmitter signal loss is a function of the distance and must accounted for during the simulation.



### **4.3.7 S-Interface Communication Component**

The S-interface component provides the BRI connectivity between the ISDN user and the ISTA. This connection is similar to most wiring configuration which can be used to connect to an NT. These configurations can be divided into three types:

- A single installation where only one terminal is connected to an NT
- A multi-terminal installation where several terminals are connected to an NT1 via a passive bus
- A multi-terminal installation where several terminals are connected to an NT1 or an NT2 in a star configuration

At the outset both the ISIS and FSIS architecture will use a single point installation between the ISDN Telephone and the VSAT. This will permit the use of up to 1000m of cable to assure maximum of 6 dB attenuation at 96 kHz. This cable length will provide a signal round trip delay of 10 to 42 microseconds from the transmitter to the receiver.

The S-interface is modeled by a single process (SIF) that delays the message as a function of the round trip delay. For both the ISIS and FSIS Network Model all protocol messages are sent on the D-channel and therefore have a constant delay once the D -channel contention has been resolved.

### **4.3.8 U-Interface Communication Component**

The U-interface component provides the transfer of information that takes place on the two wire circuit between the ISTA and the VSAT. For both the ISIS and FSIS architecture echo cancelling is used. Echo cancelling is characterized by simultaneous transmission in both direction, full duplex, elimination of echo, and a bit rate of 160 kbps. The 144 kbps are used for the 2B+D BRI information and the other 16 kbps is used for synchronization, operations, and maintenance.

The U-interface is modeled by a single process (UIF) that delays the messages as a function of its BRI rate. For both the ISIS and FSIS Network Model all protocol messages are sent on the D-channel and therefore have a constant delay.

## **4.4 ISIS and FSIS Simulation Processes**

This section describes the software simulation processes that make up the communication components of both the ISIS and FSIS Network architectures. Their use in the ISIS and FSIS end-to end simulation processes diagram depicted in Figures 4.3-5 and 4.3-2, respectively.. These processes are the software modules that implement the communication functions being modeled. As indicated above, each of these processes/modules is reused in a number of the communication components that make up the ISIS and FSIS Network Models. The same description format is used in order to provide a direct comparison between the processes

### **4.4.1 ISDN Protocol Process -- FSISP**

The ISDN Protocol Processor process accepts ISUP command messages; takes the appropriate call control action of assigning and relinquishing B-channel resources; sends appropriate ISUP status messages.

Inputs: ISUP Command Messages

Outputs: ISUP Status Messages

**Operation:**

- Accept ISUP Command Message from ISIS and FSIS Process
- Assign B-Channel
- Relinquish B-Channel
- Send ISUP Status Message
- Accommodate Design Processing Time
- Return

**Parameters:** Number of B-Channels Available for assignment  
Processing Time for each accept, assign, relinquish and send action

#### **4.4.2 Order Wire Process -- ISISO, FSISO**

The Order Wire Process accepts TDMA signals from the VSAT via the uplink receiver (Rx30) and converts them to ISDN basic access frames and routes them to the Q921 process. The protocol conversion process continues up the the Q931 and ISUP layers to the ISDN Protocol Processor (FSISP). On the other side the ISISO, FSISO and MGSO processes accepts ISDN basic access frames from the Q921 process, convert them to TDMA signals and routes them to the satellite downlink transmitter.

**Inputs:** TDMA Signals, Basic Access Frame

**Outputs:** TDMA Signals, Basic Access Frame

**Operations:**

- Accept TDMA Signals from Rx30 Process
- Convert to Basic Access Frames
- Route TDMA Signals to Q921 Process
- Accommodate Design Processing Time
- Return

Accept Basic Access Frames from ISISO and FSISO Process

- Convert to TDMA Signal
- Route TDMA Signal to Tx20 Process
- Accommodate Design Processing Time
- Return

**Parameters:** Processing Time for each accept and route action

#### **4.4.3 I430 Process**

The I430 process is based on the CCITT Recommendation I.430, Basic User Interface - Layer 1 Specification, for the point-to-point operation at Layer 1 for a single transmitter (source) and receiver (sink) are active at one time. The nominal transmitted bit rate at the interface is cited as 192 kbps in both direction of transmission. The activation/deactivation sequence shown in Figure 4.4.3-1, "Layer 1 Protocol Activation/Deactivation" will be used. The processing of associated management primitives is reserved for future implementations.

The I430 process will propagate all higher layer messages without error to and from the S-interface via the Info3 and Info4 transmissions in F7-Activated and G3-Active states.

**Inputs** PH-ACTIVATE, PH-ACTIVATE-IND, PH-DEACTIVATE IND, INFO0, INFO2, INFO3[TE-->NE], and INFO4[TE<--NE] Messages

**Outputs:** PH-ACTIVATE IND, PH-DEACTIVATE IND, INFO0, INFO2, INFO3[TE-->NE], and INFO4[TE<--NE] Messages

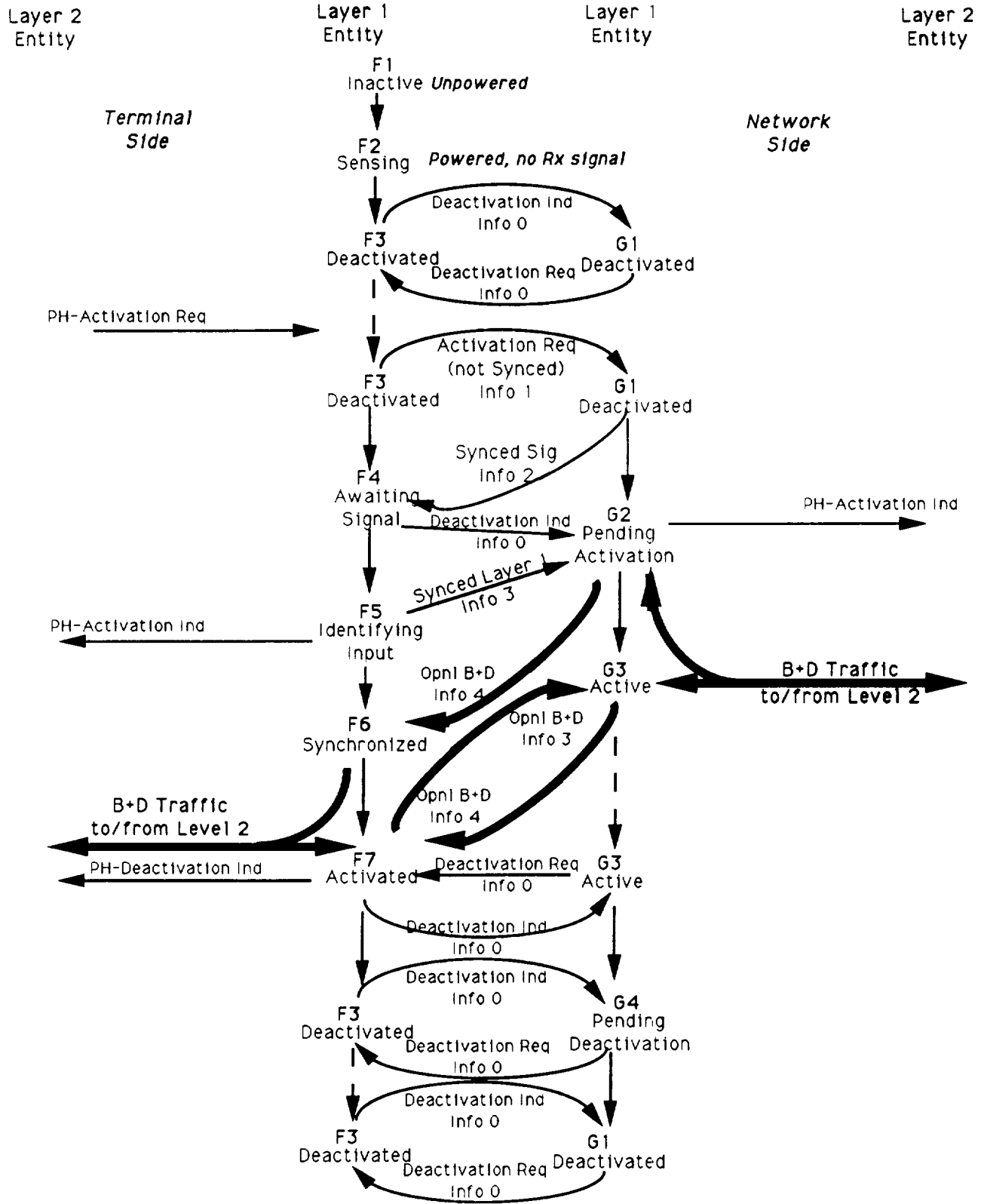


Figure 4.4.3-1 Layer 1 Protocol Activation/Deactivation

Operations:

Accept PH-ACTIVATE from Q921 Process

- if F3-Deactivated State
- Start T3-Timer
- Send Info1(non-synced) to SIF Process
- Set F4-Awaiting Signal State
- Accommodate Design Processing Time
- else Return

Accept Info1 from SIF Process

- if G3-Deactivated State
- Send Info2 (within 1sec) to SIF Process
- Start T1-Timer
- Set G2-Pending Activation State
- Accommodate Design Processing Time
- else Return

Accept Info2 from SIF Process

- if F4-Awaiting Signal State
- Send Info0 (within 5 millisecc) to SIF Process
- Set F5-Identifying Input
- Send Info3 (within 100 millisecc of Info2 to SIF Process)
- Set F6-Synchronized
- Accommodate Design Processing Time
- else Return

Accept Info3 from SIF Process

- if G3-Active State
- Send [Info3] to Q921 Process
- else if
- G3-Pending Activation State
- Send Info4 (within 500 millisecc) to SIF Process
- Send PH-ACTIVATED IND to Q921 Process
- Set G3-Active State
- Accommodate Design Processing Time
- else Return

Accept Info4 from SIF Process

- if F7-Activated State
- Send [Info4] to Q921 Process
- else if
- F6-Synchronized State
- Send PH-ACTIVATED IND to Q921 Process
- Set F7-Activated State
- Accommodate Design Processing Time
- else Return

Parameters: Values of T1, T2, and T3 timers  
Processing Time for each accept, if, and send action

#### 4.4.4 I431 Process

The I431 process is based on the CCITT Recommendation I.431. Primary Rate User-Network Interface - Layer 1 Specification, for the point-to-point operation at Layer 1 for a single transmitter (source) and receiver (sink) are active at one time. The nominal transmitted bit rate at the interface is cited as 1544 kbps in both direction of transmission. The interfaces for the primary rate user-network interface is active at all times. No activation/deactivation are applied to the interface. The F1-Operational State and the G1-Operational State are assumed to be active. The other fault condition states are left for future implementations.

Inputs: Layer 2 and UIF Frames

Outputs: Layer 2 and UIF Frames

Operations:

Accept Layer 2 Frames from UIF Process

if G1-Operational State

- Send Layer 2 Frames to Q921 Process
- Accommodate Design Processing Time
- else Return

Accept Layer 2 Frames from UIF Process

- if F1-Operational State
  - Send Layer 2 Frames to Q921 Process
  - Accommodate Design Processing Time
- else Return

Parameters: Processing Time for each accept, if, and send action

#### 4.4.5 ISDN Telephone Process -- ISDNO, ISDND

The ISDN Telephone process is based on human interface that requests and terminates ISDN telephone calls. The ISDNO process acts as a source by generating a Layer 3 protocol sequence that triggers the Q931 process. The timing and content of these initiating messages are obtained from the scenario traffic file (STF).

Inputs: Scenario Traffic File (STF).

Outputs: Request and Terminate Messages

Operations:

Read STF

- Send Q931 Message to Q931 process
- loop

Accept Q931 Message from Q931 Process

- Set ISDNO and ISDND Internal State
- Send Q931 Response Message to Q931 Process
- Accommodate Design Processing Time
- Return

Parameters: Processing Time for each accept and send action

#### 4.4.6 ISUP Process

The ISUP process provides its end-user with the capability to establish, supervise, and terminate basic bearer services. As currently defined, the ISUP is restricted to 64 kbps switched connections. The message structures and functional procedures for carrying out ISUP tasks are given in CCITT Recommendations Q.730, Q.761 to Q.764, and Q.766. For the FSIS Network Model, the ISUP functions are performed within the ISDN Satellite. For ISIS the ISUP functions are performed by the MGS.

Inputs: Q931 and ISUP Messages

Outputs: Q931 and ISUP Messages

Operations:

Accept Q931 Message from Q931 Process

- Send ISUP Message to ISISO and FSISO Process
- Accommodate Design Processing Time
- Return

Accept ISUP Message from VSATO Process

- Send Q931 Message to Q931 Process
- Accommodate Design Processing Time
- Return

Parameters: Processing Time for each accept and send action

#### 4.4.7 MGS Order Wire Process -- MGSO

The Mission Ground Station (MGS) order wire process accepts TDMA signals from the ISIS satellite downlink receiver and converts them to ISDN basic access frames and routes them to the Q921 process. The protocol conversion process continues up the the Q931 and ISUP layers to the ISDN Protocol Processor (MGSP). On the other side, the MGSO process accepts ISDN basic access frames from the Q921 process, convert them to TDMA signals and routes them to the ISIS satellite uplink transmitter (Tx30).

Inputs: TDMA Signals, Basic Access Frame

Outputs: TDMA Signals, Basic Access Frame

Operations:

Accept TDMA Signals from Rx30 Process

- Convert to Basic Access Frames
- Route TDMA Signals to Q921 Process
- Accommodate Design Processing Time
- Return

Accept Basic Access Frames from ISISO and FSISO Process

- Convert to TDMA Signal
- Route TDMA Signal to Tx20 Process or Tx30 Process
- Accommodate Design Processing Time
- Return

Parameters: Processing Time for each accept and route action

#### 4.4.8 MGS Processor Process

The Mission Ground Station (MGS) ISDN Protocol Processor process accepts ISUP command messages; takes the appropriate order wire call control action of assigning and relinquishing B-channel resources; sends appropriate ISUP status messages.

Inputs: ISUP Command Messages

Outputs: ISUP Status Messages

Operation:

Accept ISUP Command Message from MGSO Process

- Assign B-Channel
- Relinquish B-Channel
- Send ISUP Status Message
- Accommodate Design Processing Time
- Return

Parameters: Number of B-Channels Available for assignment

Processing Time for each accept, assign, relinquish and send action

#### 4.4.9 Propagation Process

The Propagation (Prop) process models all space propagation aspects for both the ISIS and FSIS Network Model. The distance between transmitter and receiver reduces the amount of energy (SigPropEnergy) available at the receiver. The weather conditions also affect the SigPropEnergy and are included in this Prop process. The notation "\*\*\*" is used to represent 20Ghz, 30Ghz, or other frequencies as appropriate.

Inputs: Any Message from TxAnt\*\* Process

Outputs: Input Message to RxAnt\*\* Process

Operations:

Accept Message from TxAnt\*\* Process

- if Communication Component (input) is VSAT
  - Adjust SigPropEnergy by Prop(VSATLoc,ACTSLoc)
  - Adjust SigPropEnergy by Prop(WeaVSAT)
  - Send Message to ISDNOW RxAnt30 Process
  - Accommodate Design Processing Time
- else Return

Accept Message from TxAnt\*\* Process

if Communication Component (input) is ISIS and FSIS  
Adjust SigPropEnergy by Prop(VSATLoc,ACTSLoc)  
Adjust SigPropEnergy by Prop(WeaVSAT)  
Send Message to VSAT RxAnt20 Process  
Accommodate Design Processing Time  
else Return

Parameters: Processing Time for each accept, if, and send action

WeaVSAT = weather between VSAT and ACTS

#### 4.4.10 Q921 Process

The Q921 process provides data link peer-to-peer exchange of information of the Link Access Procedures on the D-channel, LAPD. The CCITT Recommendation Q.921, ISDN User-Network Interface - Data Link Layer Specification provide a description of the procedures and function of LAPD. This LAPD protocol is used in the ISDNO and ISDND, ISTA, and the VSAT communication components to assure error free peer-to-peer protocol message exchanges in the D-channel.

Inputs: I430, FSISO, MGSO, and Q931 Messages

Outputs: I430, FSISO, MGSO, and Q931 Messages

Operations:

- Accept I430 Message from I430 Process
- Send Q931 Message to Q931 Process
- Accommodate Design Processing Time
- Return

- Accept FSISO and MGSO Message from FSISO and MGSO Processes
- Send FSISO and MGSO Messages to Q931 Process
- Accommodate Design Processing Time
- Return

- Accept Q931 Message from Q931 Process
- Send I430 Message to I430 Process
- Accommodate Design Processing Time
- Return

Parameters: Processing Time for each accept, if, and send action

#### 4.4.11 Q931 Process

The Q931 process provides procedures for establishing, maintaining, and clearing network connections at the ISDN user-network interface. Messages are exchanged over the D-channel. The CCITT Recommendation Q.931, ISDN User-Network Interface Layer 3 Specification for Basic Call Control provide a description of the procedures and functions. For the ISIS Network Model this Q.931 protocol is used in the ISDNO, ISDND, ISTA, and the VSAT communications component to assure error free peer-to-peer protocol message exchanges in the D channel.

Inputs: ISDNO, ISDND, ISUP, and Q931 Messages

Outputs: ISDNO, ISDND, ISUP, and Q931 Messages

Operations:

- Accept ISDNO and ISDND Messages from ISDNO and ISDND Processes
- Convert ISDNO and ISDND Messages to Q931 Message
- Send Q931 Message to Q921 Process
- Accommodate Design Processing Time
- Return

- Accept ISUP Message from ISUP Process
- Convert ISUP Message to Q931 Message
- Send Q931 to Q921 Process
- Accommodate Design Processing Time
- Return



- Accept Q931 Message from Q921 Process
  - if Communication Component is ISDNO or ISDND
    - Convert Q931 Message to ISDNO or ISDND Messages
  - Send ISDNO and ISDND Messages
    - to ISDNO and ISDND Process, respectively
    - Accommodate Design Processing Time
- else Return

Parameters: Processing Time for each accept, if, and send action

#### 4.4.12 Rx \*\* Process

The Rx\*\* process models all receivers of both the ISIS and FSIS Network Architectures. The "\*\*\*" notation is place holder for 20Ghz, 30Ghz, or other frequencies that represent the downlink and uplink frequencies of the Network Models. The receivers have a sensitivity parameter that set the energy values below which a signal is not accepted. For signal energy below the receiver sensitivity the whole message is consider loss. That message is logged to the MSave file as a lost message together with the time/subsystem that failed it. The message is not propagated.

Inputs: Any Message from RxAnt\*\*  
 Outputs: Input Message to FSISO, ISISO, MGSO or VSATO  
 Operations:

- Accept Message from RxAnt\*\* Process
  - if MsgPropValue > Rx\*\*Sensitivity
    - if Communication Component is ISDN Satellite
      - Send Message to the appropriate FSISO or ISISO Process
      - Return
  - if Communication Component is VSAT
    - Send Message to VSATO Process
    - Accommodate Design Processing Time
    - Return
  - if Communication Component is MGS
    - Send Message to MGSO Process
    - Accommodate Design Processing Time
    - Return
- else Return

Parameters: Processing Time for each accept, if, and send action  
 Rx\*\*Sensitivity as receiver threshold

#### 4.4.13 RxAnt \*\* Process

The RxAnt\*\* process models all receiver antennas of the FSIS Network Model. The "\*\*\*" notation is place holder for 20Ghz, 30Ghz, or other frequencies that represent the downlink and uplink frequencies of the ISIS or FSIS Network architectures. The receiver antennas have a number of parameters that reflect its design. RxAnt\*\*BW represents the antenna beam; RxAnt\*\*G sets the antenna gain; RxAnt\*\*Lat and RxAnt\*\*Lon indicate the antenna subpoint location; RxAnt\*\*Dwell represents the antenna dwell time at a location; RxAnt\*\*HopFreq represents its hop frequency; and RxAnt\*\*Scan provides the antenna

scan rate. To be received by the corresponding receiver the transmitted energy must coincide with all these antenna parameters.

Inputs: Any Message from Prop Process

Outputs: Input Message to Rx\*\* Process

Operations:

Accept Message from Prop Process - Case

- if Communication Component is VSAT
  - if VSAT RxAnt20 pointed at ACTS Satellite
    - Adjust SigPropEnergy by RxAnt20Gain
    - Send Message to VSAT Rx20 Process
    - Accommodate Design Processing Time
  - else Return
- if Communication Component is ISDN Satellite
  - if ISDN Satellite RxAnt30 pointed at transmitter subpoint
    - Adjust SigPropEnergy by RxAnt30Gain
    - Send Message to ACTS Rx30 Process
    - Accommodate Design Processing Time
  - else Return

if Communication Component is VSAT

- if VSAT RxAnt20 pointed at ACTS Satellite
  - Adjust SigPropEnergy by RxAnt20Gain
  - Send Message to VSAT Rx20 Process
  - Accommodate Design Processing Time
- else Return

default Return

Parameters: Processing Time for each accept, if, and send action

RxAnt\*\*BW = antenna beam

RxAnt\*\*Gain = antenna gain

RxAnt\*\*Lat = antenna latitude

RxAnt\*\*Lon = antenna longitude

RxAnt\*\*Dwell = antenna dwell time

RxAnt\*\*HopFreq = antenna hop frequency

RxAnt\*\*Scan = antenna scan rate.

#### 4.4.14 SIF Process

The SIF process models the S-interface between the user ISDN Telephone and the ISDN Satellite Terminal Adapter (ISTA). For the FSIS Network Model, the SIF Process provides basic rate ISDN (BRI) connectivity for the I.430 Basic Access Frames.

Inputs: I430 Protocol Basic Access Frames from the I430 Process

Outputs: I430 Protocol Basic Access Frames to the I430 Process

Operations:

Accept I430 Basic Access Frames from I430 Process

- Send I430 Basic Access Frames to I430 Process
- Accommodate Design Processing Time
- Return

Parameters: Processing Time for each accept and send action

#### 4.4.15 Tx \*\* Process

The Tx\*\* process models all transmitters of the FSIS Network Model. The "\*\*\*" notation is place holder for 20Ghz, 30Ghz or other frequencies that represent the ISIS and FSIS downlink and uplink frequencies. The transmitters are modeled as isotropic radiators that added to the signal being transmitted with SigPropEnergy value. This value is mitigated by the TxAnt\*\*, propagation (Prop) and RxAnt\*\* processes, and finally used by the Rx\*\* process to accept the message. .

Inputs: Signals from ISISO, FSISO, MGSO or VSATO Process

Outputs: Signal to TxAnt\*\*

Operations:

Accept Message from MGSO or VSATO Process

- Set SigPropEnergy for Message
- Send Signal to TxAnt\*\* Process
- Accommodate Design Processing Time
- Return

Parameters: Processing Time for each accept, if, and send action  
SigPropEnergy for energy associated with message

#### 4.4.16 TxAnt \*\* Process

The TxAnt\*\* process models all transmitter antennas of the FSIS network Model. The "\*\*\*" notation is place holder for 20Ghz, 30Ghz or other frequencies that represent the ISIS and FSIS downlink and uplink frequencies. The transmitter antennas have a number of parameters that reflect its design. TxAnt\*\*BW represents the antenna beam; TxAnt\*\*Gain sets the antenna gain; TxAnt\*\*Lat and TxAnt\*\*Lon indicate the antenna subpoint location; TxAnt\*\*Dwell represents the antenna dwell time at a location; TxAnt\*\*HopFreq represents its hop frequency; and TxAnt\*\*Scan provides the antenna scan rate. To be received by the corresponding receiver antenna, the transmitted antenna energy must coincide with all these antenna parameters.

Inputs: Any Signal from Tx\*\* Process

Outputs: Input Signal to Prop Process

Operations:

Accept Signal from Tx\*\* Process

- Check RxAnt\*\* and TxAnt\*\* coincidence
- Adjust SigPropEnergy by TxAnt\*\*Gain
- Send Signal to Prop Process
- Accommodate Design Processing Time
- Return

Parameters: Processing Time for each accept, check, and send action

TxAnt\*\*BW = antenna beam

TxAnt\*\*Gain = antenna gain

TxAnt\*\*Lat = antenna latitude

TxAnt\*\*Lon = antenna longitude

TxAnt\*\*Dwell = antenna dwell time

TxAnt\*\*HopFreq = antenna hop frequency

TxAnt\*\*Scan = antenna scan rate.

#### **4.4.17 UIF Process**

The UIF process models the U-interface between the ISDN Satellite Terminal Adapter (ISTA) and the VSAT. For the FSIS Network Model, the UIF process provides primary rate ISDN (PRI) connectivity for the I431 signals.

Inputs: I431 Protocol Signal from the I431 Process

Outputs: I431 Protocol Signal to the I431 Process

Operations:

Accept I431 Signal from I431 Process

- Send I431 Signal to I431 Process
- Accommodate Design Processing Time
- Return

Parameters: Processing Time for each accept and send action

#### **4.4.18 VSAT Order Wire Process**

The VSAT Order Wire process accepts ISDN Basic Access Frame and converts them to TDMA Signal and conversely.

Inputs: ISUP and OBOW Signals

Outputs: Basic Access Frames and TDMA Signals

Operations:

Accept Basic Access Frames from I431 Process

- Convert the frame to TDMA Signal
- Sends the TDMA Signal to Tx30 Process
- Accommodate Design Processing Time
- Return

Accept TDMA Signal from Rx20 Process

- Converts TDMA Signal to a Basic Access Frame
- Sends the frame to I431 Process
- Accommodate Design Processing Time
- Return

Parameters: Processing Time for each accept, convert, and send action

## **SECTION 5**

### **SUMMARY**

#### **5.1 General**

This Simulator Design task completion report presented the complete end-to-end protocol architecture for both ISIS and FSIS suitable for discrete event simulation. The simulation processes are applicable to both the the Interim Service ISDN Satellite (ISIS) and the Full Service ISDN Satellite (FSIS). The ultimate aim of this aspect of the SCAR Program is the design of a new advanced ISDN communications satellite. The technical and operational parameters for this ISDN advanced communications satellite design will be obtained from an engineering software model of the major subsystems of the ISDN communications satellite architecture. Discrete event simulation experiments will be performed with these ISIS and FSIS models using various traffic scenarios, technical parameters, and operational procedures. The data from these simulations will be analyzed using the performance measures discussed in previous NASA SCAR reports.

#### **5.2 Review**

After an introduction that provided the background and scope of this NASA SCAR Program, the use of modeling and simulation to determine the parameters for the advanced ISDN communications satellite design was presented. An overview of the modeling and simulation tasks included a brief description of the four software programs for the effort. Section 3 discussed the SCAR simulation development and methodology used to determine the design parameters for the SCAR advanced ISDN communications satellite design. Particular attention was given to the comparison between the ISIS and FSIS simulations since over 80% of the software modules will be reused between them. The two main sections of this task completion report are Section 4.3, Simulation Structures, which include a descriptions of both ISIS and FSIS architectures down to the process level and Section 4.4, Simulation Processes, which provides a detail description of each process.

Several appendices are included to provide more detail on the Scenario Traffic File (STF), Process Array Structure, the Traffic Model Database, the Q.931 Protocol Simulation, and the Measurement Save (MSave) file.

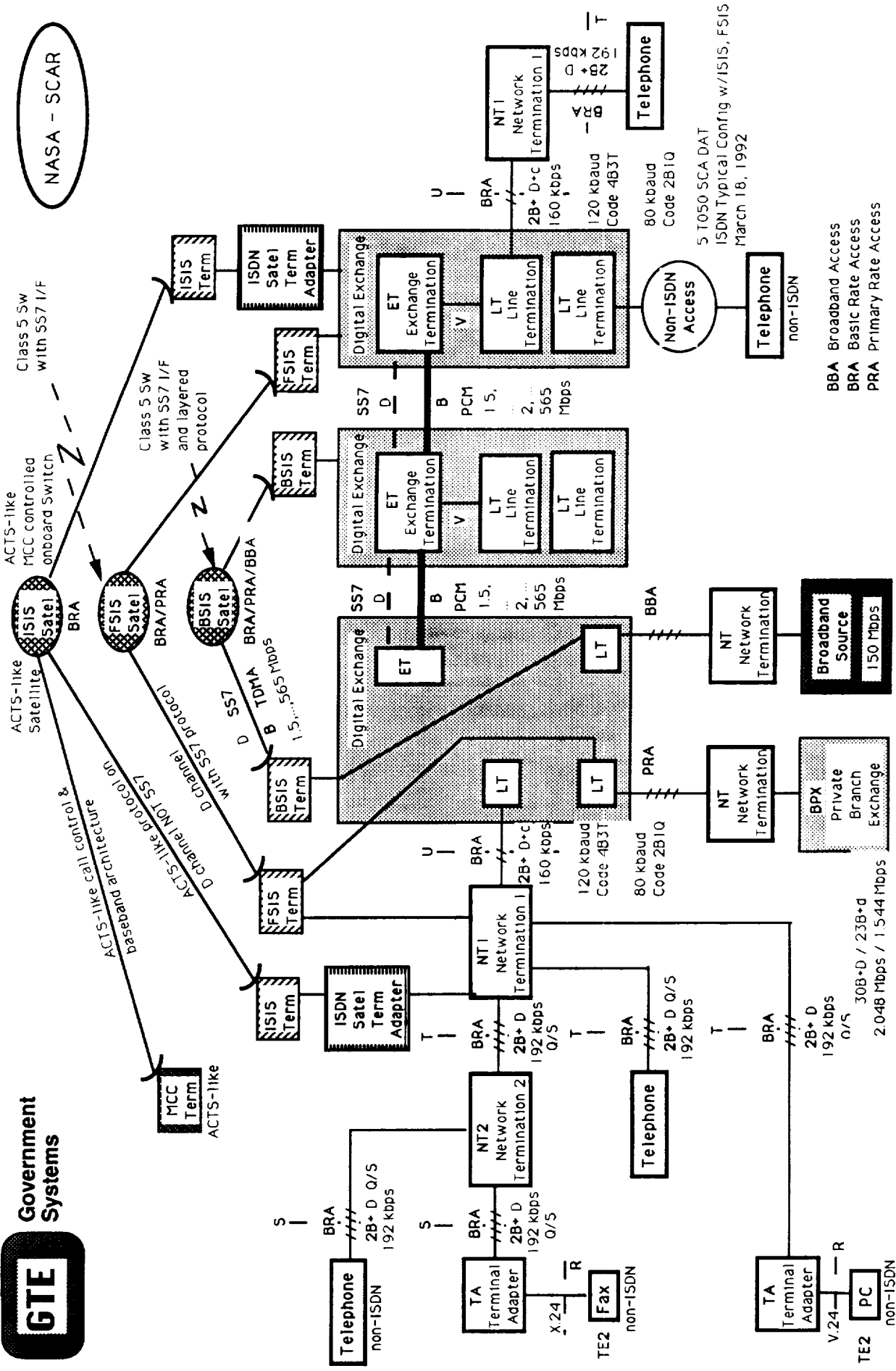
#### **5.3 Continuing Efforts**

The research in the simulator design task has been satisfactorily completed and the results are capable of supporting the NASA SCAR Program. The implementation of the ISIS and FSIS Network architectures into SIMSCRIPT II.5 will no doubt result in some refinements but the basic design will remain intact. As shown in Figure 5.3-1, "Typical ISDN Configuration with ISIS, FSIS, and BSIS Overlay", the ultimate goal is to assure that ISDN communication satellites are a viable component option for networks of the future.



Government Systems

NASA - SCAR



BBA Broadband Access  
 BRA Basic Rate Access  
 PRA Primary Rate Access

Figure 5.3-1 Typical ISDN Configuration with ISIS, FSIS and BSIS Overlay

## **Appendix A**

### **Scenario Traffic File (STF) Structure**

A section of the a STF is provided in this appendix.

The STF is a sequence of records that reflects the actions of an input scenario. Each STF record represents an event in the scenario sequence. The STF, then, is a time ordered list of events that is presented to the discrete event simulation as initiating events for a particular scenario. This list is read by the SIMSCRIPT II.5 discrete event simulation program by an external process. The external process, is called STF, activates the initiating protocol requester, ISDNO - ISDN Originator. That protocol is passed from process to process being analyzed and acted upon along the way.

Even though only three fields are visible in the STF presented in this appendix, four fields actually make up each record of the STF. The first field "STF" is used by SIMSCRIPT II.5 to identify the external file it belongs to. Since process STF is the external process reading this file, STF must head each record.

The second field "1601." of the first record is a real variable representing the simulation time in seconds. The action of the external process STF is to activate process ISDNO at that time as part of the simulation.

The next six characters after the decimal point form the Call Reference Number for that event record. The Call Reference Number uniquely identifies the service being requested and is used for every action concerning that call. The 16 format blanks leading zeroes and the following field, which is also 16, blends into the Call Reference Number making seem like a single number. The first 16 records in the STF presented in this appendix have the following Call Reference Numbers:

94, 94, 95, 95, 96, 96, 97, 97, 98, 98, 99, 99,1,1,100,100

The last field of the STF record is a combination of integer subfields. The field:

115224

is decomposed into the following information elements:

- 1 represents the Operation being requested.  
In this case 1 stands for "Rqst".  
Whereas, a 2 in that position would mean "Term"

1 represent the bearer service being requested.  
1 means a single B-channel.

52 represents the originating city identification  
with scenario generation. 52 = Washington DC

24 represents the destination city identification  
with scenario generation. 24 = Los Angeles

The first record of the STF is decoded as follows:

Request a single B-channel from Washington DC to Los Angeles  
at 1601 seconds-of-day ( 26min 41sec after mid-night)  
using Call Reference Number #94.

The second record of the STF is decoded as follows:

Terminate the service associate  
with Call Reference Number #94.  
at 1860 seconds-of-day ( 31min 00sec after mid-night).

The result is that a B-Channel was allocated to Call Reference #94 for 4  
minutes and 19 seconds.



STF#IPC.DAT

STF	1601.	94115224*
STF	1860.	94200000*
STF	3201.	95115252*
STF	3377.	95200000*
STF	4801.	96115224*
STF	5075.	96200000*
STF	6401.	97115252*
STF	6690.	97200000*
STF	8001.	98115224*
STF	8138.	98200000*
STF	9601.	99115252*
STF	9758.	99200000*
STF	11111.	1112452*
STF	11159.	1200000*
STF	11201.	100115224*
STF	11421.	2112424*
STF	11429.	100200000*
STF	11642.	2200000*
STF	11731.	3112452*
STF	11880.	3200000*
STF	12041.	4112424*
STF	12178.	4200000*
STF	12351.	5112452*
STF	12627.	5200000*
STF	12661.	6112424*
STF	12801.	101115252*
STF	12971.	7112452*
STF	12979.	101200000*
STF	13026.	6200000*
STF	13152.	7200000*
STF	13281.	8112424*
STF	13504.	8200000*
STF	13591.	9112452*
STF	13832.	9200000*
STF	13901.	10112424*
STF	14102.	10200000*
STF	14211.	11112452*
STF	14401.	102115224*
STF	14506.	11200000*
STF	14521.	12112424*
STF	14653.	102200000*
STF	14694.	12200000*
STF	14831.	13112452*
STF	14956.	13200000*
STF	15141.	14112424*
STF	15303.	14200000*
STF	15451.	15112452*
STF	15491.	15200000*
STF	15761.	16112424*
STF	15854.	16200000*
STF	16001.	103115252*
STF	16071.	17112452*
STF	16162.	17200000*
STF	16246.	103200000*

## **Appendix B**

### **Process Array (ProcAr) Structure**

A copy of a ProcAr is provided in this appendix.

The ProcAr is a matrix of engineering data that determines all the parameters associated with the discrete event simulation. The labels associated with ProcAr are shown in the table, "Process Array, Routing Table and Engineering Parameters". The actual ProcAr, engineering parameters, used for FSIS Build 1 are shown in the next two matrices. Each row of the ProcAr represents parameters associated with the Process Index (PI) in the first element of that row. Up to 9 parameters can be associated with each PI.

Since PI's represent a diversity of communication functions, such protocol analysis, receiving, transmitting, allocating, each row consists of a diversity of technical parameters. For protocol analysis, such parameters as timer values, state number are important. For the receiving function, such things as receiver threshold, noise figures, BER are meaningful. For the transmitting function, power gain and transmitting frequency play an important part. In essence, each record has parameters that pertain explicitly to that PI.

For example, only one software module is used for the receiving function, Rx. But that receiving function takes on the characteristic provided by the ProcAr record for the PI position it is modeling. It will use difference values for a 20GHz receiver than for a 30GHz receiver. Similar ProcAr technical parameters for the power generating capabilities of a space based transmitter and a ground based transmitter will be determined by the PI position they play in the simulation.

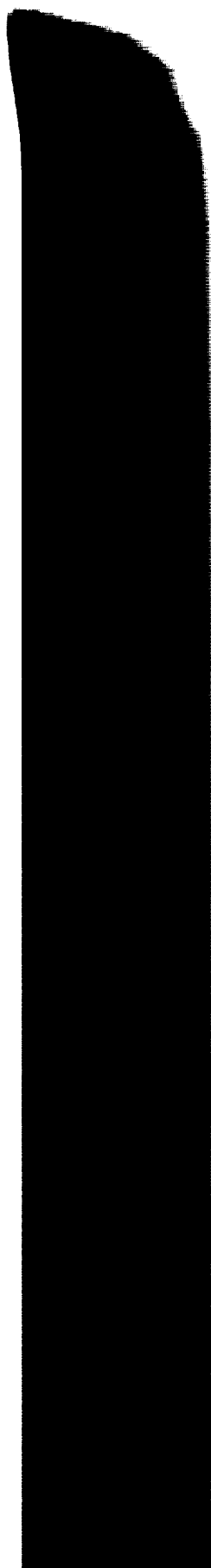
In the ProcAr Matrix in this appendix, the first column of the ten columns contains the PI number. The second column shows the delay time, in milliseconds, required for that PI to perform all its tasks. Most PI values have temporarily, arbitrarily set to 10 msec. Of particular interest is the value 6 msec/1000 miles for PI's: 13, 27, 51, and 65. Those PI's use the propagation process that multiplies this value by the number of miles. That value has temporarily, arbitrarily set at 22,000 miles resulting in a constant propagation delay of 132 msec.

The other engineering parameters, especially the timer values associated actual protocol standards for Q.931, Q.921, I430, I431, and ISUP will be added to the appropriate PI's to study their effects on end-to-end ISDN communication satellite.

### Process Array, Routing Table and Engineering Parameters

ProcAr(PI,J)		Engineering Parameters						
Process	Process Name *	*Delay (sec)	3	4	5	6		
Index (PI)	J=1	2						
1	77	ISDN Tel	##	STF Driven: Rqst and Term				
2	76	Q931	##		*T301	*T302	*T303 ... *T322	
3	75	Q921	##	*T200	*T201	*T202	*T203	
4	74	I430	##	*T1	*T2	*T3		
5	73	SIF	##	*Rate(Kbps)::D/16				
6	72	I430	##	*T1	*T2	*T3		
7	71	I431	##	*Rate(Kbps)::D/16				
8	70	UIF	##	*Rate(Kbps)::D/16				
9	69	I431	##	*Rate(Kbps)::D/16				
10	68	VSATOW	##					
11	67	TX30	##	*Freq(GHz) *Power(Db)				
12	66	TXANT30	##	*Freq(GHz) *Power(Db) *BW(deg) *HopRate(msec)				
13	65	PROP	##	*LossRate(Dd/d)				
14	64	RXANT30	##	*Freq(GHz) *Power(Db) *BW(deg) *HopRate(msec)				
15	63	RX30	##	*Freq(GHz) *Thresh(Db)				
16	62	FSISO	##					
17	61	Q921	##	*T200	*T201	*T202	*T203	
18	60	Q931	##		*T301	*T302	*T303 ... *T322	
19	59	ISUP	##					
20	58	FSISP	##	*Freq(MHz) *Mem(Mby) *WordSize(#)				
21	57	ISUP	##					
22	56	Q931	##		*T301	*T302	*T303 ... *T322	
23	55	Q921	##	*T200	*T201	*T202	*T203	
24	54	FSISO	##					
25	53	TX20	##	*Freq(GHz) *Power(Db)				
26	52	TXANT20	##	*Freq(GHz) *Power(Db) *BW(deg) *HopRate(msec)				
27	51	PROP	##	*LossRate(Dd/d)				
28	50	RXANT20	##	*Freq(GHz) *Power(Db) *BW(deg) *HopRate(msec)				
29	49	RX20	##	*Freq(GHz) *Thresh(Db)				
30	48	VSATOW	##					
31	47	I431	##	*Rate(Kbps)::D/16				
32	46	UIF	##	*Rate(Kbps)::D/16				
33	45	I431	##	*Rate(Kbps)::D/16				
34	44	I430	##	*T1	*T2	*T3		
35	43	SIF	##	*Rate(Kbps)::D/16				
36	42	I430	##	*T1	*T2	*T3		
37	41	Q921	##	*T200	*T201	*T202	*T203	
38	40	Q931	##		*T301	*T302	*T303 ... *T322	
39		ISDN Tel	##	Protocol Response; Always available				
78		ISDN Sat		*Lat(deg)	*Lon(deg)	*Alt(Kmi)		
79		B-Channels		BChMax	BChAvail			

1	10	0	0	0	0	0	0	0	0
2	10	0	0	0	0	0	0	0	0
3	10	0	0	0	0	0	0	0	0
4	10	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0
	10	0	0	0	0	0	0	0	0



1	10	0	0	0	0	0	0	0	0
2	10	0	0	0	0	0	0	0	0
3	10	0	0	0	0	0	0	0	0
4	10	0	0	0	0	0	0	0	0
5	10	0	0	0	0	0	0	0	0
6	10	0	0	0	0	0	0	0	0
7	10	0	0	0	0	0	0	0	0
8	10	0	0	0	0	0	0	0	0
9	10	0	0	0	0	0	0	0	0
10	10	0	0	0	0	0	0	0	0
11	20	30	100	0	0	0	0	0	0
12	5	30	8000	10	20	0	0	0	0
13	6	0	-100	0	0	0	0	0	0
14	5	30	9000	40	20	0	0	0	0
15	30	30	100	0	0	0	0	0	0
16	10	0	0	0	0	0	0	0	0
17	10	0	0	0	0	0	0	0	0
18	10	0	0	0	0	0	0	0	0
19	10	0	0	0	0	0	0	0	0
20	10	0	2	0	0	0	0	0	0
21	10	0	0	0	0	0	0	0	0
22	10	0	0	0	0	0	0	0	0
23	10	0	0	0	0	0	0	0	0
24	10	0	0	0	0	0	0	0	0
25	10	30	100	0	0	0	0	0	0
26	5	30	8000	10	20	0	0	0	0
27	6	0	-100	0	0	0	0	0	0
28	5	30	9000	40	20	0	0	0	0
29	30	30	100	0	0	0	0	0	0
30	10	0	0	0	0	0	0	0	0
31	10	0	0	0	0	0	0	0	0
32	10	0	0	0	0	0	0	0	0
33	10	0	0	0	0	0	0	0	0
34	10	0	0	0	0	0	0	0	0
35	10	0	0	0	0	0	0	0	0
36	10	0	0	0	0	0	0	0	0
37	10	0	0	0	0	0	0	0	0
38	10	0	0	0	0	0	0	0	0
39	10	0	0	0	0	0	0	0	0
40	10	0	0	0	0	0	0	0	0
41	10	0	0	0	0	0	0	0	0
42	10	0	0	0	0	0	0	0	0
43	10	0	0	0	0	0	0	0	0
44	10	0	0	0	0	0	0	0	0
45	10	0	0	0	0	0	0	0	0
46	10	0	0	0	0	0	0	0	0
47	10	0	0	0	0	0	0	0	0
48	10	0	0	0	0	0	0	0	0
49	20	30	100	0	0	0	0	0	0
50	5	30	8000	10	20	0	0	0	0
51	6	0	-100	0	0	0	0	0	0
52	5	30	9000	40	20	0	0	0	0
53	30	30	100	0	0	0	0	0	0
54	10	0	0	0	0	0	0	0	0

55	10	0	0	0	0	0	0	0	0
56	10	0	0	0	0	0	0	0	0
57	10	0	0	0	0	0	0	0	0
58	10	0	2	0	0	0	0	0	0
59	10	0	0	0	0	0	0	0	0
60	10	0	0	0	0	0	0	0	0
61	10	0	0	0	0	0	0	0	0
62	10	0	0	0	0	0	0	0	0
63	10	30	100	0	0	0	0	0	0
64	5	30	8000	10	20	0	0	0	0
65	6	0	-100	0	0	0	0	0	0
66	5	30	9000	40	20	0	0	0	0
67	30	30	100	0	0	0	0	0	0
68	10	0	0	0	0	0	0	0	0
69	10	0	0	0	0	0	0	0	0
70	10	0	0	0	0	0	0	0	0
71	10	0	0	0	0	0	0	0	0
72	10	0	0	0	0	0	0	0	0
73	10	0	0	0	0	0	0	0	0
74	10	0	0	0	0	0	0	0	0
75	10	0	0	0	0	0	0	0	0
76	10	0	0	0	0	0	0	0	0
77	10	0	0	0	0	0	0	0	0
78	5	0	-100	0	0	0	0	0	0
79	0	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0	0
81	0	0	0	0	0	0	0	0	0
82	0	0	0	0	0	0	0	0	0
83	0	0	0	0	0	0	0	0	0
84	0	0	0	0	0	0	0	0	0
85	0	0	0	0	0	0	0	0	0
86	0	0	0	0	0	0	0	0	0
87	0	0	0	0	0	0	0	0	0
88	0	0	0	0	0	0	0	0	0
89	0	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0	0
91	0	0	0	0	0	0	0	0	0
92	0	0	0	0	0	0	0	0	0
93	0	0	0	0	0	0	0	0	0
94	0	0	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0	0
96	0	0	0	0	0	0	0	0	0
97	0	0	0	0	0	0	0	0	0
98	0	0	0	0	0	0	0	0	0
99	0	0	0	0	0	0	0	0	0
100	0	0	0	0	0	0	0	0	0

## **Appendix C**

### **Traffic Model Database Data**

This Traffic Model consists of a number of databases: the City Reference DB, the ISDN User vs Industry DB, the Application vs Industry DB, the Application vs Time DB, and the Application vs Bearer Services DB. The "City Reference Database", DB1, identifies the percentage of ISDN users that are associated with the population of fifty-four major cities. Due to paucity of specific ISDN user information this percentage factor will be used as multiplier of population to infer the number of ISDN users in that region. The geographic coordinates of these of these cities together with their US time-zone are included in the Traffic Model in order to provide a sub-point for communications satellite operations. A view of the geographical distribution of these CONUS Traffic Model Cities is shown in "NASA/SCAR Traffic Model".

The "ISDN User vs Industry", DB2, apportions the ISDN traffic among twenty-one industries. These data permit the scenario selection on an industry-by-industry basis. This database is used in conjunction with the City Reference Database to further decompose the ISDN service use in terms of industry affiliation. The "Application vs Industry Database", DB3, further apportions the industry into applications of communication services. This added data granularity permits the selection of scenarios tailored on an application basis. The nine applications are spread across each of the twenty-one industries on a percentage basis too permit each application to contribute in a normalized fashion.

The "Applications vs Time Database", DB4, associates daily time-slots for issuing ISDN service requests on an application basis. These data allows the generation of traffic distributions that are appropriate to the application being used in a scenario. The hours in a day are divided into four unequal time slots along the line of a typical work day: 0001-0800, 0801-1200, 1201-1800, and 1801-2400. The "Application vs ISDN Bearer Service, Message Length Database", DB5, associates ISDN bearer services with the selected scenario applications. For this SCAR program the following ISDN bearer services have been selected: circuit switched (64 kbps and 128 kbps), D-Channel X.25, B-Channel Frame Relay, and Telemetry. This database also associates the message length and message hold-time with each application. These message duration values provide a measure of the length of time each ISDN bearer service is used.

### City Reference Database

SCAR Database I.

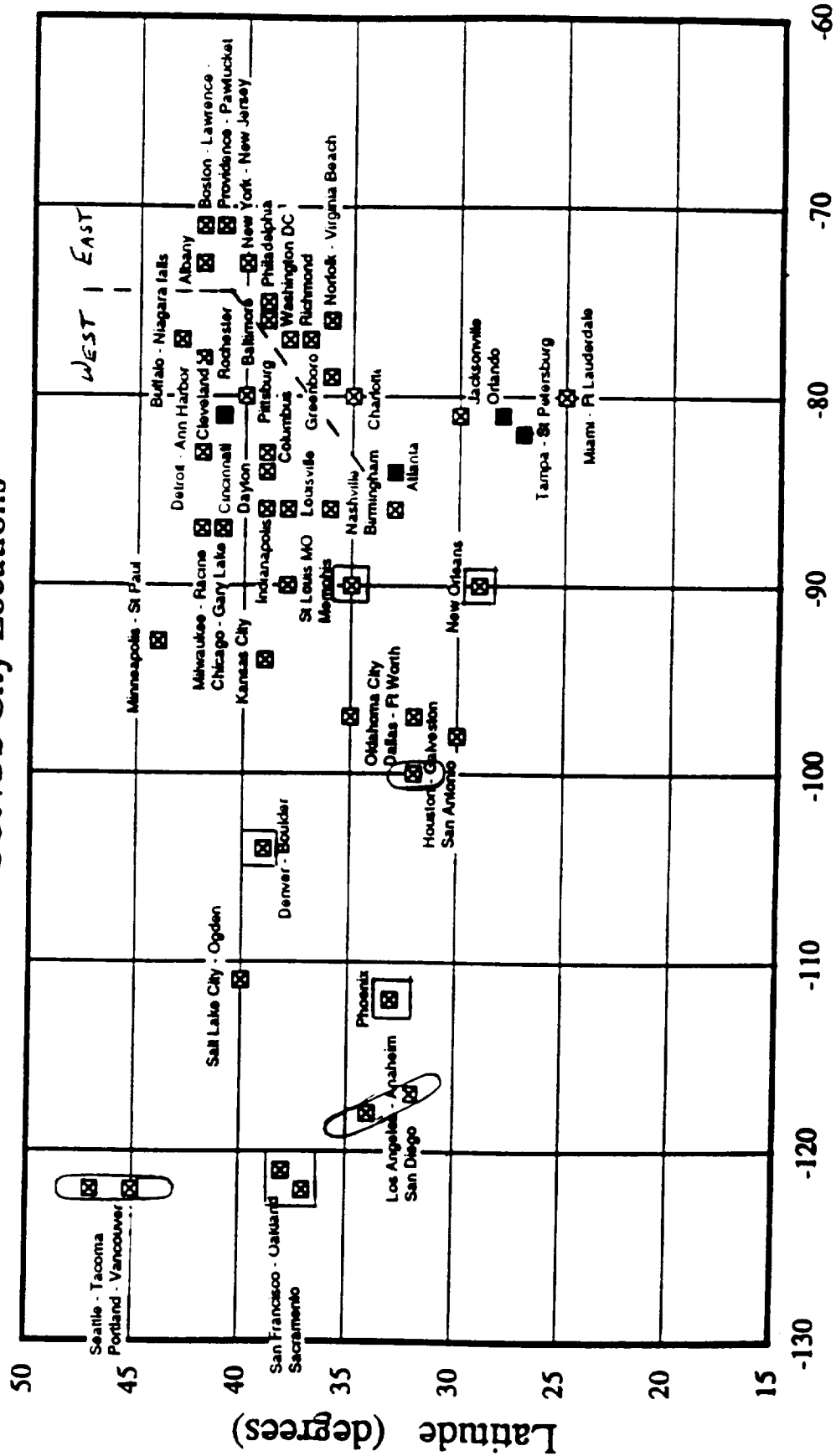
CITYNAME	POPULATION ,000	LATITUDE		ISDNPCT	
		deg	deg	%	TIMEZONE
Honolulu	838	21	-157	3.30	-5
Anchorage	227	61	-150	3.10	-4
Seattle-Tacoma	2421	47	-122	3.40	-3
Portland-Vancouver	1414	45	-122	3.10	-3
San Francisco-Oakland-San Jose	6042	37	-122	4.00	-3
Sacramento	1385	38	-121	3.30	-3
Los Angeles-Anaheim-Riverside	13770	34	-118	4.50	-3
San Diego	2370	32	-117	3.30	-3
Phoenix	2030	33	-112	3.30	-2
Salt Lake City-Ogden	1065	40	-111	3.10	-2
Denver-Boulder	1858	39	-103	3.10	-2
Houston-Galveston	3641	32	-100	3.40	-1
San Antonio	1323	30	-98	3.10	-1
Oklahoma City	964	35	-97	3.20	-1
Dallas-Fort Worth	3766	32	-97	3.40	-1
Kansas City	1575	39	-94	3.10	-1
Minneapolis-St. Paul	2388	44	-93	3.30	-1
St. Louis	2467	38	-90	3.20	-1
Memphis	979	35	-90	3.10	-1
New Orleans	1307	29	-90	3.10	-1
Milwaukee-Racine	1572	42	-87	3.10	-1
Chicago-Gary Lake County	8181	41	-87	3.90	0
Indianapolis	1237	39	-86	3.10	0
Nashville	972	36	-86	3.10	-1
Birmingham	923	33	-86	3.10	-1
Louisville	967	38	-85	3.10	0
Cincinnati-Hamilton	1728	39	-84	3.20	0
Dayton-Springfield	948	39	-84	3.20	0
Atlanta	2737	33	-84	3.20	0
Detroit-Ann Arbor	4620	42	-83	3.30	0
Columbus	1344	39	-83	3.10	0
Tampa-St. Petersburg-Clearwater	1995	27	-82	3.20	0
Cleveland-Akron-Lorain	2769	41	-81	3.30	0
Jacksonville	898	30	-81	3.10	0
Orlando	971	28	-81	3.20	0
Pittsburgh-Beaver Valley	2284	40	-80	3.20	0
Charlotte-Gastonia-Rocky Hill	1112	35	-80	3.10	0
Miami-Fort-Lauderdale	3001	25	-80	3.30	0
Greensboro-Winston-Salem-High	925	36	-79	3.10	0
Buffalo-Niagara Falls	1176	42	-78	3.20	0
Rochester	980	43	-77	3.20	0
Washington	3734	38	-77	3.30	0
Richmond-Petersburg	844	37	-77	3.20	0
Baltimore	2342	39	-76	3.20	0
Philadelphia-Wilmington-Trenton	5963	39	-75	3.80	0
Norfolk-Virginia Beach-Newport News	1380	36	-74	3.20	0
Hartford-New Britain-Middleton	1068	42	-73	3.00	0
Albany-Schenectady-Troy	851	42	-73	3.20	0
New York-New Jersey-Long Island	18120	40	-73	5.00	0
Boston-Lawrence-Salem	4110	42	-71	3.30	0
Providence-Pawtucket-Fall River	1125	41	-71	3.00	0
San Juan-Caguas-Ponce, PR	550	18	-66	3.20	1

54 Count



# NASA/SCAR Traffic Model

CONUS City Locations



Longitude (degrees)

**ISDN User vs Industry***SCAR Database 2.*

<i>Industry</i>	<i>ISDN</i> %
BROADCAST	4.0
COMMUNICATION	10.0
CONSTRUCTION	2.0
DATA PROCESSING	2.0
EDUCATION	6.0
ENERGY	2.0
FINANCIAL	8.0
FOOD SERVICE	2.0
GOVERNMENT	8.0
LEGAL	6.0
LODGING	4.0
MANUFACTURING	6.0
MEDICAL	6.0
MILITARY	10.0
PUBLISHING	4.0
RECREATION	4.0
RESIDENTIAL	2.0
RETAIL	4.0
TRANSPORT	6.0
UTILITY	2.0
WHOLESALE	2.0
---	-- Check
21 Count	100.0 Normalization



## Application vs Time Database

### SCAR Database 4

APPLICATION	0001-0800	0801-1200	1201-1800	1801-2400	Check Normalization
	mid/8am 8 hours TIMENO1 %	8am/noon 4 hours TIMENO2 %	noon/6pm 6 hours TIMENO3 %	6pm/mid 6 hours TIMENO4 %	
Voice(interactive)	2.5	32.0	51.0	14.5	100.0
Voice(message)	2.5	32.0	51.0	14.5	100.0
Facsimile	2.5	32.0	51.0	14.5	100.0
FileTransfer	52.0	3.0	5.0	40.0	100.0
VideoBroadcasting	10.0	25.0	30.0	35.0	100.0
VideoConference	2.5	32.0	51.0	14.5	100.0
InteractiveData	2.5	32.0	51.0	14.5	100.0
Transaction	2.5	32.0	51.0	14.5	100.0
Teletex	2.5	32.0	51.0	14.5	100.0

## Application vs ISDN Bearer Service, Message Length Database

SCAR Database 5.

APPLICATION	CS64KBPS		CS128KBPS		DX25		BFRAMERELY		TELEMETRY		Check Normalization		Message HOLDTIME		Message Length kbytes
	%		%		%		%		%		%		min		
Voice(interactive)	100.0		0.0		0.0		0.0		0.0		0.0		100.0	3.0	0
Voice(message)	100.0		0.0		0.0		0.0		0.0		0.0		100.0	1.5	0
Facsimile	80.0		15.0		2.0		3.0		0.0		0.0		100.0	0.0	160
File Transfer	30.0		20.0		30.0		15.0		5.0		5.0		100.0	0.0	216
VideoBroad	25.0		40.0		0.0		35.0		0.0		0.0		100.0	0.0	9600
VideoConf	30.0		40.0		0.0		30.0		0.0		0.0		100.0	0.0	57600
Interactivedata	20.0		10.0		40.0		10.0		20.0		20.0		100.0	0.0	27
Transaction	20.0		20.0		40.0		10.0		10.0		10.0		100.0	0.0	27
Teletex	10.0		10.0		40.0		20.0		20.0		20.0		100.0	0.0	75

## Appendix D

### Q.931 Protocol Simulation

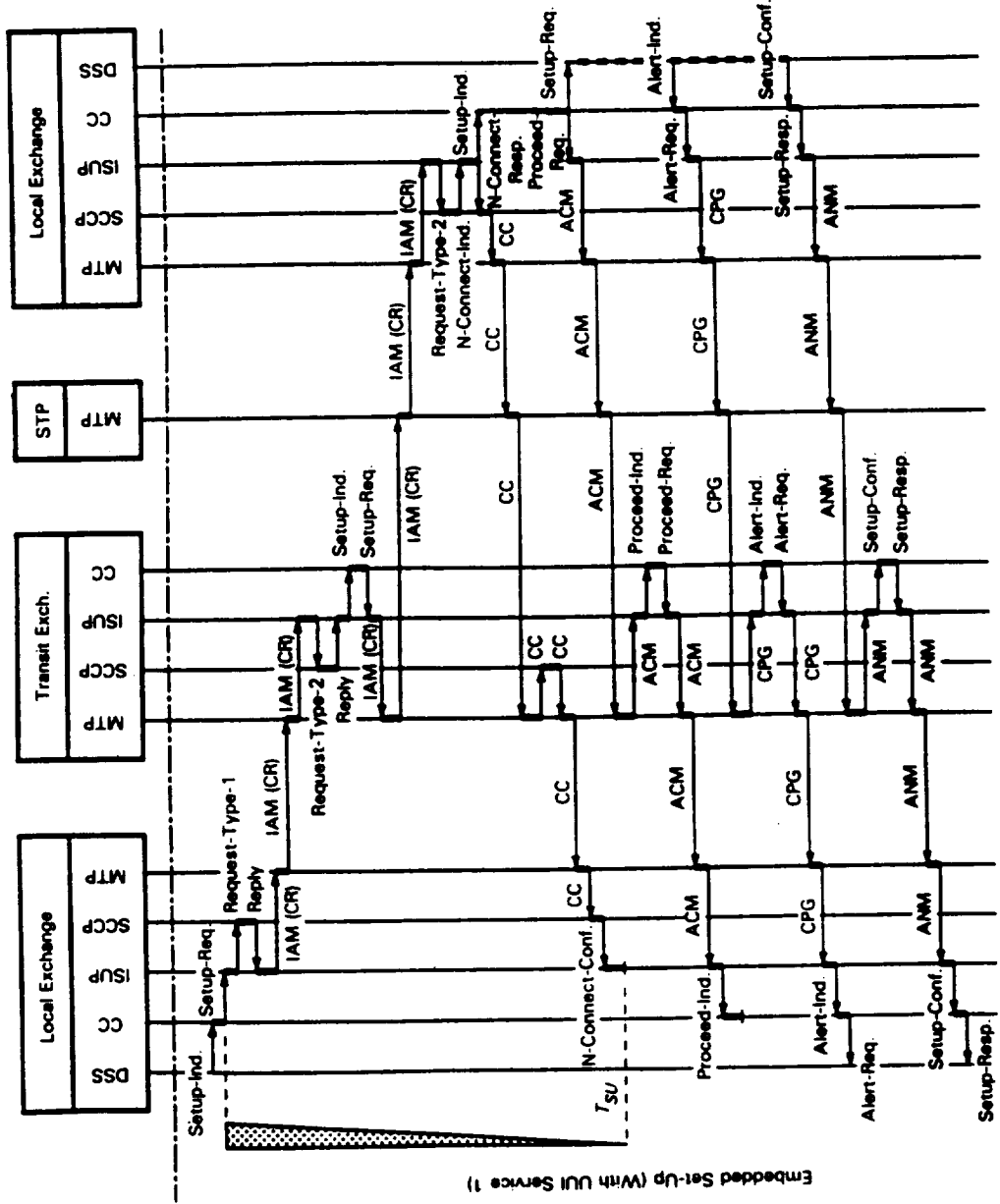
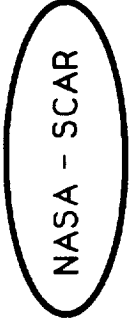
The Q931 process provides procedures for establishing, maintaining, and clearing network connections at the ISDN user-network interface. Messages are exchanged over the D-channel. The CCITT Recommendation Q.931, ISDN User-Network Interface Layer 3 Specification for Basic Call Control provide a description of the procedures and functions.

This protocol like all other protocol will be simulated at the message element level. All aspects of the protocol necessary set up an ISDN service, as depicted in the "Part of Unidirectional ISDN Call Scenario" will implemented. The protocols depicted by the Specification Design Language (SDL) diagrams will be converted to a matrix depicting the initial state, the intervening message, and the output state. In our example, the "Part of the SDL Diagram for Outgoing Call Setup Procedure" can be traced to seven record entries in the "Q.931 Network-side States" matrix. The top level of the SDL diagram in this appendix is converted to the third record in the Q931 matrix. In this way the many pages of SDL diagrams can be reduce to a single matrix that can now be accessed by the corresponding Q931 process. A view of both the user states and the network states for a preliminary version of the Q931 "Software Flow Diagram" process implementation is shown in the last pages of this appendix. That process description look more intimidating than it actual is. Less than ten line of code are needed to access the data in either matrix. The principal task is centered at reducing the many SDL diagrams to a single matrix.



Government Systems

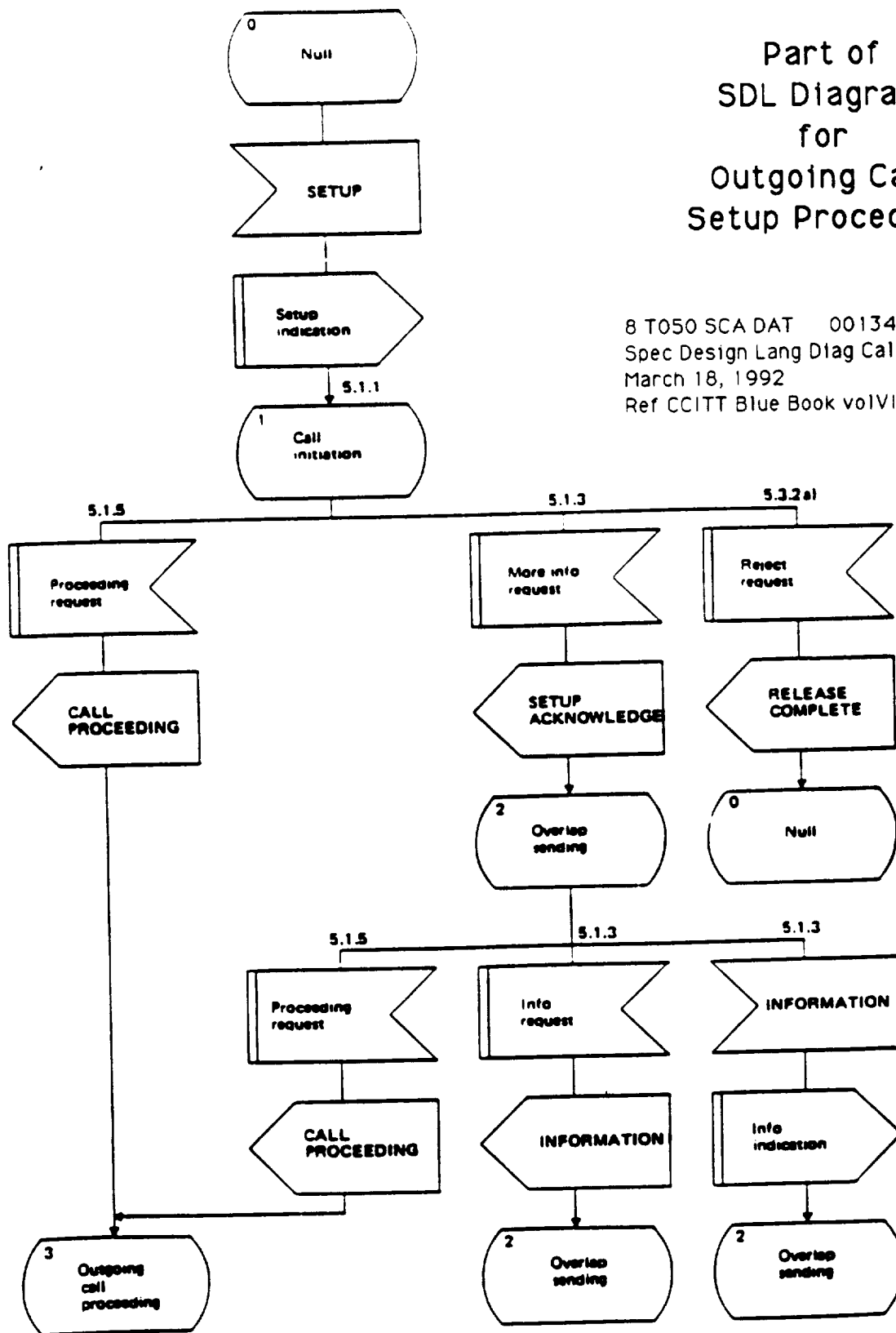
Part of Unidirectional ISDN Call Scenario



Embedded Set-Up (With UI Service 1)

Part of  
SDL Diagram  
for  
Outgoing Call  
Setup Procedure

8 T050 SCA DAT 00134-00  
Spec Design Lang Diag Call Setup Proc  
March 18, 1992  
Ref CCITT Blue Book volVI.11 Rec Q.931





**Q.931 Network-side States**

<i>Initial State</i>	<i>Primitive from Call Control</i>	<i>Message from User</i>	<i>Message to User</i>	<i>Primitive to Call Control</i>	<i>New State</i>
N00	SetUpRqst		SETUP		N06
N00		RESUME		ResumeInd	N17
N00		SETUP		SetUpInd	N01
N01	ProcRqst		CALL_PROC		N03
N01		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N01	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N01	RejRqst		RELEASE_COMP		N00
N01	MoreInfoRqst		SETUP_ACK		N02
N02		INFORMATION		InfoInd	N02
N02		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N02	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N02	InfoRqst		INFORMATION		N02
N02	ProcRqst		CALL_PROC		N03
N03	AlertRqst		ALERTING		N04
N03	ProcRqst		PROGRESS		N03
N03		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N03	SetUpResp		CONNECT		N10
N03	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N04		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N04	SetUpResp		CONNECT		N10
N04	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N06		CONNECT		SetUpConf	N08
N06		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N06		RELEASE_COMP		ReleaseInd	N00
N06		ALERTING		AlertInd	N07
N06	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N06		CALL_PROC		Proclnd	N09
N07		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N07	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N07		CONNECT		SetUpConf	N08
N08		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N08	SetUpCompRqst		CONN_ACK		N10
N08	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N09		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N09		PROGRESS		ProgressInd	N09
N09		CONNECT		SetUpConf	N08
N09	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N09		ALERTING		AlertInd	N07
N10		SUSPEND		SuspInd	N15
N10	NotifyRqst		NOTIFY		N10
N11		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N11	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N11	ReleaseRqst				N19
N12		DISCONNECT	RELEASE		N19
N12		RELEASE	<b>[ReleaseBCh]</b>	ReleaseInd	N00
N15		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N15	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N15	SuspRejRqst		SUSP_REJ		N10
N15	SuspResp		SUSP_ACK		N00
N17	DisconnRqst	<b>[DisconnBCh]</b>	DISCONNECT		N12
N17	ResumeRejRqst		RESUME_REJ		N10
N17	ResumeResp		RESUME_ACK		N10
N17		DISCONNECT	<b>[DisconnBCh]</b>	DisconnInd	N11
N19		RELEASE_COMP	<b>[ReleaseBCh]</b>	ReleaseInd	N00

<i>15</i> <i>Initial State</i>	<i>14</i> <i>Primitive from Call Control</i>	<i>11</i> <i>Message from User</i>	<i>16</i> <i>Message to User</i>	<i>9</i> <i>Primitive to Call Control</i>	<i>15</i> <i>New State</i>
-----------------------------------	---	---------------------------------------	-------------------------------------	--	-------------------------------

### Software Flow Diagram

Input Parameters:

CallRef

PI

Msg

where:

PI :: ProcessIndex

*Resume call*

Predefined Parameter Definitions, ProcAr(L):

Param Name

Param Units

Q931Delay(PI)

sec



TIME.V, CallRef, PI, Msg, StatusIndex :: Entry  
(MSave)

Calculate ISDNTElDelay  
{ CallRef, ProcAr(PI) }  
Wait ISDNTElDelay

Operation

Calculate next PI  
Determine next Process  
\* = CallRef,NewPI,...

Decision

FSIS

< PI= 2,38,41,76 :: User side? > -y->

UserState

Go to CallRefAr(CallRef,PI)

CallRefAr(L)

ProcessState

SecState

ProcessState	SecState	Action
0 ->	6	Msg=S -> ~Next Process w/*SI~ -> end
0 ->	17	Msg=RR -> ~Next Process w/*R~ -> end
0 ->	1	Msg=SR -> ~Next Process w/*S~ -> end
1 ->	3	Msg=CP -> ~Next Process w/*PI~ -> end
1 ->	2	Msg=SA -> ~Next Process w/*MII~ -> end
2 ->	12	Msg=D -> ~Next Process w/*DI~ -> end
2 ->	0	Msg=R -> ~Next Process w/*RI+RC~ -> end
2 ->	3	Msg=CP -> ~Next Process w/*PI~ -> end
2 ->	11	Msg=DR -> ~Next Process w/*D~ -> end
2 ->	2	Msg=IR -> ~Next Process w/*I~ -> end
3 ->	4	Msg=A -> ~Next Process w/*AI~ -> end
3 ->	10	Msg=C -> ~Next Process w/*SC~ -> end
3 ->	12	Msg=D -> ~Next Process w/*DI~ -> end
3 ->	0	Msg=R -> ~Next Process w/*RI+RC~ -> end
3 ->	11	Msg=DR -> ~Next Process w/*D~ -> end
4 ->	12	Msg=D -> ~Next Process w/*DI~ -> end
4 ->	0	Msg=R -> ~Next Process w/*RI+RC~ -> end
4 ->	11	Msg=R -> ~Next Process w/*RI+RC~ -> end
5 ->		-> [MSave] with TIME.V, CallRef, PI, Msg, StatusIndex :: U05 Err
6 ->	7	Msg=AR -> ~Next Process w/*A~ -> end
6 ->	25	Msg=MIR -> ~Next Process w/*SA~ -> end
6 ->	9	Msg=PR -> ~Next Process w/*CP~ -> end
6 ->	0	Msg=Re R -> ~Next Process w/*RC~ -> end
6 ->	8	Msg=SR -> ~Next Process w/*C~ -> end
7 ->	12	Msg=D -> ~Next Process w/*DI~ -> end
7 ->	0	Msg=R -> ~Next Process w/*RI+RC~ -> end
7 ->	11	Msg=DR -> ~Next Process w/*D~ -> end
7 ->	8	Msg=SR -> ~Next Process w/*C~ -> end
8 ->	10	Msg=CA -> ~Next Process w/*SCI~ -> end
8 ->	12	Msg=D -> ~Next Process w/*DI~ -> end
8 ->	0	Msg=R -> ~Next Process w/*RI+RC~ -> end
8 ->	11	Msg=DR -> ~Next Process w/*D~ -> end
9 ->	12	Msg=D -> ~Next Process w/*DI~ -> end
9 ->	0	Msg=R -> ~Next Process w/*RI+RC~ -> end
9 ->	7	Msg=AR -> ~Next Process w/*A~ -> end
9 ->	11	Msg=DR -> ~Next Process w/*D~ -> end
9 ->	8	Msg=SR -> ~Next Process w/*C~ -> end
10 ->	12	Msg=D -> ~Next Process w/*DI~ -> end
10 ->	0	Msg=R -> ~Next Process w/*RI+RC~ -> end
10 ->	10	Msg=N -> ~Next Process w/*NI~ -> end
10 ->	11	Msg=DR -> ~Next Process w/*D~ -> end
10 ->	15	Msg=SR -> ~Next Process w/*S~ -> end
11 ->	19	Msg=D -> ~Next Process w/*R~ -> end
11 ->	0	Msg=R -> ~Next Process w/*RI+RC~ -> end
12 ->	0	Msg=R -> ~Next Process w/*RI+RC~ -> end
12 ->	19	Msg=RR -> ~Next Process w/*R~ -> end
13 ->		-> [MSave] with TIME.V, CallRef, PI, Msg, StatusIndex :: U13 Err
14 ->		-> [MSave] with TIME.V, CallRef, PI, Msg, StatusIndex :: U14 Err
15 ->	0	Msg=SA -> ~Next Process w/*SC~ -> end
15 ->	10	Msg=SR -> ~Next Process w/*SRI~ -> end
16 ->		-> [MSave] with TIME.V, CallRef, PI, Msg, StatusIndex :: U14 Err
17 ->	10	Msg=RA -> ~Next Process w/*RC~ -> end
17 ->	10	Msg=RR R -> ~Next Process w/*RRI~ -> end
18 ->		-> [MSave] with TIME.V, CallRef, PI, Msg, StatusIndex :: U14 Err
19 ->	0	Msg=RC -> ~Next Process w/*RC~ -> end
25 ->	12	Msg=D -> ~Next Process w/*DI~ -> end
25 ->	25	Msg=I -> ~Next Process w/*II~ -> end
25 ->	0	Msg=R -> ~Next Process w/*RI+RC~ -> end
25 ->	7	Msg=AR -> ~Next Process w/*A~ -> end
25 ->	11	Msg=DR -> ~Next Process w/*D~ -> end
25 ->	9	Msg=PR -> ~Next Process w/*CP~ -> end
25 ->	8	Msg=SR -> ~Next Process w/*C~ -> end

Decision

< PI= 18,22,56,61 :: Network side? > ---

-y-->

Go to CallRefAr(CallRef,PI)

CallRefAr(U)

PresentState

SetState

0 -->	6	Msg=S --> ~Next Process w/^SI~ --> end
0 -->	17	Msg=RR --> ~Next Process w/^R~ --> end
0 -->	1	Msg=SR --> ~Next Process w/^S~ --> end
1 -->	3	Msg=PR --> ~Next Process w/^CP~ --> end
1 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
1 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
1 -->	0	Msg=RejR --> ~Next Process w/^RC~ --> end
1 -->	2	Msg=MIR --> ~Next Process w/^SA~ --> end
2 -->	2	Msg=l --> ~Next Process w/^l~ --> end
2 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
2 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
2 -->	2	Msg=iR --> ~Next Process w/^i~ --> end
2 -->	3	Msg=PR --> ~Next Process w/^CP~ --> end
3 -->	4	Msg=AR --> ~Next Process w/^A~ --> end
3 -->	3	Msg=PR --> ~Next Process w/^P~ --> end
3 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
3 -->	10	Msg=SR --> ~Next Process w/^C~ --> end
3 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
4 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
4 -->	10	Msg=SR --> ~Next Process w/^C~ --> end
4 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
5 -->		--> [MSave] with TIME.V, CallRef, PI, Msg, StatusIndex :: N05 Err
6 -->	8	Msg=C --> ~Next Process w/^SC~ --> end
6 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
6 -->	0	Msg=RC --> ~Next Process w/^RI~ --> end
6 -->	7	Msg=A --> ~Next Process w/^AI~ --> end
6 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
6 -->	9	Msg=CP --> ~Next Process w/^PI~ --> end
7 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
7 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
7 -->	8	Msg=C --> ~Next Process w/^SC~ --> end
8 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
8 -->	10	Msg=SCR --> ~Next Process w/^CA~ --> end
8 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
9 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
9 -->	9	Msg=P --> ~Next Process w/^PI~ --> end
9 -->	8	Msg=C --> ~Next Process w/^SC~ --> end
9 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
9 -->	7	Msg=A --> ~Next Process w/^AI~ --> end
10 -->	15	Msg=S --> ~Next Process w/^SI~ --> end
10 -->	10	Msg=NR --> ~Next Process w/^N~ --> end
11 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
11 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
11 -->	19	Msg=R --> ~Next Process w/^RI?~ --> end
12 -->	0	Msg=R --> ~Next Process w/^RI~ --> end [Release B-Ch]
12 -->	19	Msg=D --> ~Next Process w/^R~ --> end
13 -->		--> [MSave] with TIME.V, CallRef, PI, Msg, StatusIndex :: N13 Err
14 -->		--> [MSave] with TIME.V, CallRef, PI, Msg, StatusIndex :: N14 Err
15 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
15 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
15 -->	10	Msg=SRejR --> ~Next Process w/^SR~ --> end
15 -->	0	Msg=SR --> ~Next Process w/^SA~ --> end
16 -->		--> [MSave] with TIME.V, CallRef, PI, Msg, StatusIndex :: N16 Err
17 -->	12	Msg=DR --> ~Next Process w/^D~ --> end [Disconn B-Ch]
17 -->	10	Msg=RRRejR --> ~Next Process w/^RRRej~ --> end
17 -->	10	Msg=RRResp --> ~Next Process w/^RA~ --> end
17 -->	11	Msg=D --> ~Next Process w/^DI~ --> end [Disconn B-Ch]
18 -->		--> [MSave] with TIME.V, CallRef, PI, Msg, StatusIndex :: N18 Err
19 -->	0	Msg=RC --> ~Next Process w/^RC~ --> end [Release B-Ch]

[ MSave ]

TIME.V, CallRef, I, Msg, StatusIndex :: PI\*10 (PI error)

end

## **Appendix E**

### **Measurement Save (MSave) Structure**

A copy of one of the MSave files generated by the FSIS Build 1 is provided in this appendix.

The Measurement Save (MSave) contains the time sequenced data generated by the simulation. Every time a communication process is activated an entry is made in MSave. Also, all significant events, such as call connection, disconnection and blocking are saved in MSave. Periodically the internal MSave array is saved to disk as MSAVE##.DAT. Where ## represents a sequential digit starting with 1. These MSave files are the principal data sources for post simulation analyses. They contain all the data that can be obtained from a particular simulation.

The MSave contains data that is relevant to call processing in an ISDN era. It is expected that the data within MSave can be used to calculate throughput, response time, blocking probability, and robustness. By using these performance measures an advanced ISDN satellite can be postulated and tested using these simulation techniques.

The MSave record consists of five fields: the simulation time, the call reference number, the PI, the protocol msg, and a status indicator. For FSIS Build 1, the protocol message was replaced by the combination in the STF to show the Operation/B-channel/Orig City/Dest City.

For example the first record of the MSave file in the appendix shows the simulation time in milliseconds-of-day:

12661634 milliseconds after midnight

The second field is call reference number associated with this call:

6 is the Call Reference Number

The third field identifies the number of the PI generating the record:

41 is the PI

For FSIS Build 1, the fourth field contains the last entry in the STF that originated the event. As mentioned previously:

112424 represents:

1 :: Rqst  
1 :: B-Channel bearer service  
24 :: Originating City (Los Angeles)  
24 :: Destination City (Los Angeles)

The fifth field represents the status of the Call Reference at the time indicated.  
For FSIS Build 1 these status definitions apply:

1 Process entry  
2 Process exit  
3 Call Connect  
4 Call Disconnect  
5 Call Blocked

Using the combination of Call Reference Number and Status index the progression of a call can be traced as it enters each process (PI) and the corresponding actions take for that call.

**For example:** Call Reference #101 requested a B-Channel at 12801.000 secs. That request progressed through 58 PI's until it was connected at 12801.996 secs. Call #101 requested a disconnect at 12979.000 secs; that disconnect request was affected at 12979.342 secs.

The analysis:           12801.000           Rqst           (Status = 1)  
                          12801.996           Connected   (Status = 3)  
                          therefore Connect Response Time: .996 secs

                          12979.000           Term         (Status = 1)  
                          12979.342           Disconnected (Status = 4)  
                          therefore Disconnect Response Time: .342 secs

Call Reference #101 used the B-Channel between  
                          12979.342  
                          12801.996

-----  
                          177.346 secs (2min 57.3sec)

Note. Call Reference #7 requested a B-Channel at 12971.000. It was declared blocked (Status Index = 5) at 12971.996.

12661674	6	41112424	1
12661684	6	42112424	1
12661694	6	43112424	1
12661704	6	44112424	1
12661714	6	45112424	1
12661724	6	46112424	1
12661734	6	47112424	1
12661744	6	48112424	1
12661754	6	49112424	1
12661774	6	50112424	1
12661779	6	51112424	1
12661911	6	52112424	1
12661916	6	53112424	1
12661946	6	54112424	1
12661956	6	55112424	1
12661966	6	56112424	1
12661976	6	57112424	1
12661986	6	58112424	1
12661996	6	58112424	3
12801000	101	1115252	1
12801010	101	2115252	1
12801020	101	3115252	1
12801030	101	4115252	1
12801040	101	5115252	1
12801050	101	6115252	1
12801060	101	7115252	1
12801070	101	8115252	1
12801080	101	9115252	1
12801090	101	10115252	1
12801100	101	11115252	1
12801120	101	12115252	1
12801125	101	13115252	1
12801257	101	14115252	1
12801262	101	15115252	1
12801292	101	16115252	1
12801302	101	17115252	1
12801312	101	18115252	1
12801322	101	19115252	1
12801332	101	20115252	1
12801342	101	21115252	1
12801352	101	22115252	1
12801362	101	23115252	1
12801372	101	24115252	1
12801382	101	25115252	1
12801392	101	26115252	1
12801397	101	27115252	1
12801529	101	28115252	1
12801534	101	29115252	1
12801564	101	30115252	1
12801574	101	31115252	1
12801584	101	32115252	1
12801594	101	33115252	1
12801604	101	34115252	1
12801614	101	35115252	1

**NASA**National Aeronautics and  
Space Administration

## Report Documentation Page

1. Report No.		2. Government Accession No:		3. Recipient's Catalog No.	
4. Title and Subtitle Simulator Design for Advanced Satellite Designs and Experiments (Task Completion Report)				5. Report Date 30 March 1992	
7. Author(s) Gerard R. Pepin				6. Performing Organization Code	
9. Performing Organization Name and Address GTE Government Systems 15000 Conference Center Drive P.O. Box 10814 Chantilly, VA 22021-3808				8. Performing Organization Report No.	
12. Sponsoring Agency Name and Address NASA Headquarters Headquarters Acquisition Division 300 7th Street, SW Washington, DC 20546-0001				10. Work Unit No.	
				11. Contract or Grant No. NASW-4520	
				13. Type of Report and Period Covered Simulator Design Task Completion Report (25 SEP 90 - 1 MAR 92)	
				14. Sponsoring Agency Code	
15. Supplementary Notes					
16. Abstract This Simulation Design Task Completion Report documents the simulation techniques associated with the network models of both the Interim Service ISDN Satellite (ISIS) and the Full Service ISDN Satellite (FSIS) architectures. The ISIS Network Model design represents satellite systems like the Advanced Communications Technology Satellite (ACTS) orbiting switch.  The FSIS architecture, the ultimate aim of this element of the SCAR Program, moves all control and switching functions on-board the next generation ISDN communications satellite. The technical and operational parameters for the advanced ISDN communications satellite design will be obtained from the simulation of ISIS and FSIS engineering software models for their major subsystems. Discrete event simulation experiments will be performed with these models using various traffic scenarios, design parameters, and operational procedures. The data from these simulations will be used to determine the engineering parameters for the advanced ISDN communications satellite.					
17. Key Words (Suggested by Author(s)) ISDN, satellite, traffic network, simulation, ISDN standards, B-ISDN, frame relay, on-orbit switching, computer networks, satellite orbits, satellite transmission quality, network configuration, traffic model			18. Distribution Statement Unclassified-Unlimited		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of pages	22. Price