DECEMBER 1985

MDC H1343A

SSDS

MDAC

Ford IBM RCA

SPACE STATION DATA SYSTEM ANALYSIS/ARCHITECTURE STUDY

Task 2 – Options Development, DR-5
Volume II – Design Options

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY

MCDONNELL DOUGLAS
CORPORATION

MCDONNELL DOUGLAS CORPORATION

SPACE STATION DATA SYSTEM ANALYSIS/ARCHITECTURE STUDY

Task 2 – Options Development, DR-5
Volume II – Design Options

DECEMBER 1985

MDC H1343A
REPLACES MDC H1940
DATED MAY 1985

MCDONNELL DOUGLAS ASTRONAUTICS COMPANY-HUNTINGTON BEACH
5301 Bolsa Avenue Huntington Beach, California 92647 (714) 896-3311

PREFACE

The McDonnell Douglas Astronautics Company has been engaged in a Space Station Data System Analysis/Architecture Study for the National Aeronautics and Space Administration, Goddard Space Flight Center. This study, which emphasized a system engineering design for a complete, end-to-end data system, was divided into six tasks:

Task 1. Functional Requirements Definition
Task 2. Options Development
Task 3. Trade Studies
Task 4. System Definitions
Task 5. Program Plan
Task 6. Study Maintenance

McDonnell Douglas was assisted by the Ford Aerospace and Communications Corporation, IBM Federal Systems Division and RCA in these Tasks. The Task inter-relationship and documentation flow are shown in Figure 1.

This report was prepared for the National Aeronautics and Space Administration Goddard Space Flight Center under Contract No. NAS5-28082

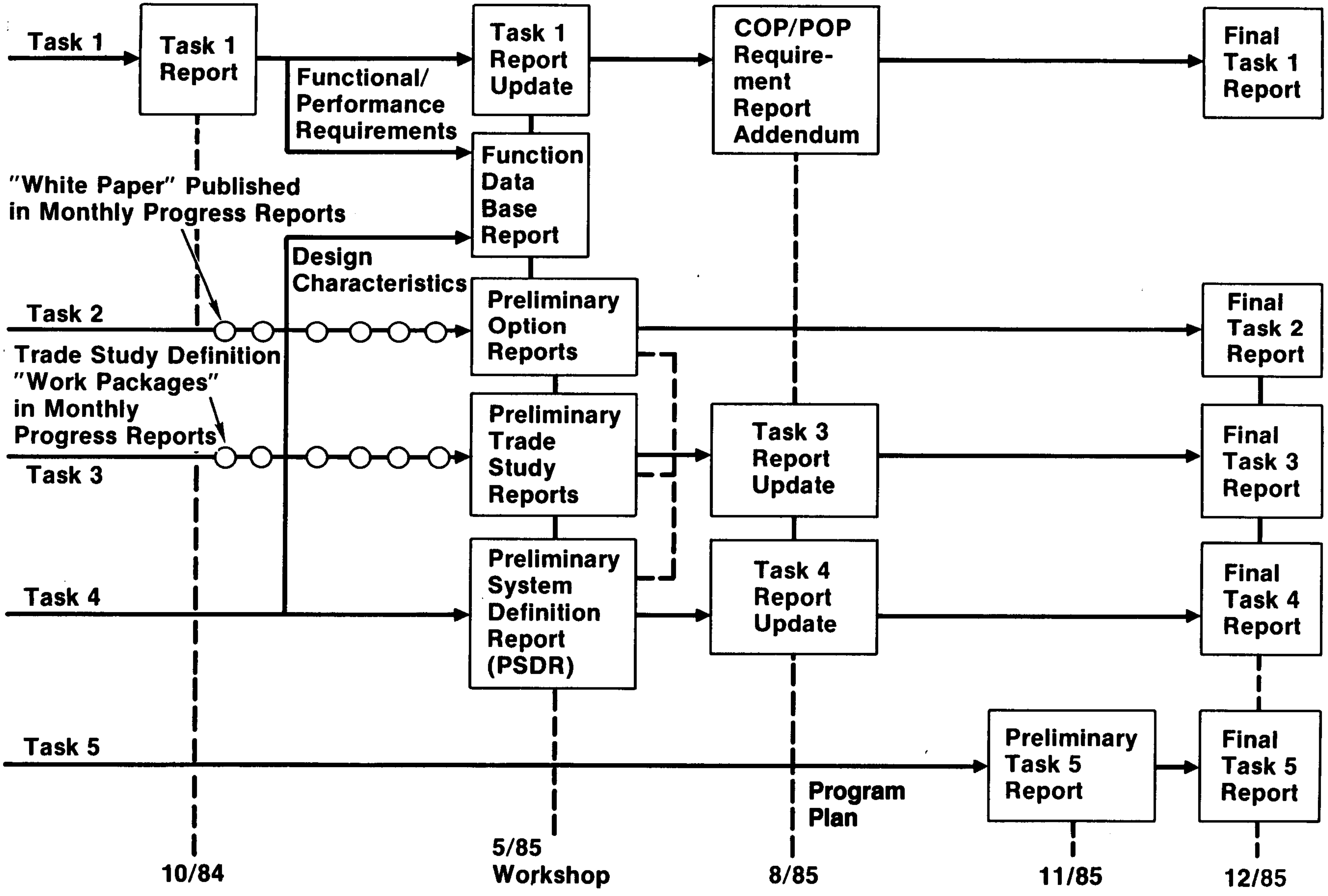
Questions regarding this report should be directed to:

Glen P. Love
Study Manager
McDonnell Douglas Astronautics Company
Huntington Beach, CA 92647
(714) 896-2292

Figure 1

VHG598

SSDS A/A DOCUMENTATION SCHEDULE



Task 1

Task 1 Report

Functional/ Performance Requirements

Task 1 Report Update

Function Data Base Report

COP/POP Require-ment Report Addendum

Final Task 1 Report

"White Paper" Published in Monthly Progress Reports

Design Characteristics

Task 2

Preliminary Option Reports

Final Task 2 Report

Trade Study Definition "Work Packages" in Monthly Progress Reports

Task 3

Preliminary Trade Study Reports

Task 3 Report Update

Final Task 3 Report

Task 4

Preliminary System Definition Report (PSDR)

Task 4 Report Update

Final Task 4 Report

Task 5

Program Plan

Preliminary Task 5 Report

Final Task 5 Report

10/84

5/85 Workshop

8/85

11/85

12/85

Task 2, Volume II

TABLE OF CONTENTS

TOPIC PAGE

TOPIC PAGE

GLOSSARY

A	Automatic
A&R	Automation and Robotics
A/A	Analysis/Architecture
A/D	Advanced Development
A/L	Airlock
A/N	Alphanumeric
AC&S	Attitude Control System
ACA	Attitude Control Assembly
ACO	Administrative Contracting Officer
ACS	Attitude Control and Stabilization
ACS/COM	Attitude Control System/Communications
ACTS	Advanced Communications Technology Satellite
AD	Ancillary Data
AD	Advanced Development
ADOP	Advanced Distributed Onboard Processor
ADP	Advanced Development Plan
AFOSR	Air Force Office of Scientific Research
AFP	Advanced Flexible Processor
AFRPL	Air Force Rocket Propulsion Laboratory
AGC	Automatic Gain Control
AGE	Attempt to Generalize
AI	Artificial Intelligence
AIE	Ada Integrated Environment
AIPS	Advanced Information Processing System
AL1	Air Lock One
ALS	Alternate Landing Site
ALS/N	Ada Language System/Navy
AMIC	Automated Management Information Center
ANSI	American National Standards Institute
AOS	Acquisition of Signal
AP	Automatic Programming
APD	Avalanche Photo Diode
APSE	Ada Programming Support Environment
ARC	Ames Research Center

ART	Automated Reasoning Tool
ASCII	American Standard Code for Information Exchange
ASE	Airborne Support Equipment
ASTROS	Advanced Star/Target Reference Optical Sensor
ATAC	Advanced Technology Advisory Committee
ATC	Air Traffic Control
ATP	Authority to Proceed
ATPS	Advanced Telemetry Processing System
ATS	Assembly Truss and Structure
AVMI	Automated Visual Maintenance Information
AWSI	Adoptive Wafer Scale Integration
B	Bridge
BARC	Block Adaptive Rate Controlled
BB	Breadboard
BER	Bit Error Rate
BIT	Built-in Test
BITE	Built-in Test Equipment
BIU	Buffer Interface Unit
BIU	Bus Interface Unit
BIU	Built-in Unit
BMD	Ballistic Missile Defense
BTU	British Thermal Unit
BW	Bandwidth
C	Constrained
C^2	Command and Control
C^3	Command, Control, and Communication
C^3I	Command, Control, Communication, and Intelligence
C&DH	Communications and Data Handling
C&T	Communication and Tracking Subsystem
C&T	Communications and Tracking
C&W	Control and Warning
C/L	Checklist
CA	Customer Accommodation
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAIS	Common APSE Interface Set
CAM	Computer-Aided Manufacturing

CAMAC	Computer Automatic Measurement and Control
CAP	Crew Activities Plan
CASB	Cost Accounting Standard Board
CASE	Common Application Service Elements
CATL	Controlled Aceptance Test Library
CBD	Commerce Business Daily
CBEMA	Computer and Business Equipment Manufacturing Association
CCA	Cluster Coding Algorithm
CCB	Contractor Control Board
CCB	Configuration Control Board
CCC	Change and Configuration Control
CCD	Charge-Coupled Device
CCITT	Consultive Committee for International Telegraph and Telephone
CCITT	Coordinating Committee for International Telephony and Telegraphy
CCMS	Checkout Control and Monitor System
CCR	Configuration Change Request
CCSDS	Consultative Committee for Space Data System
CCTV	Closed-Circuit Television
cd/M^2	Candelas per square Meter
CDG	Concept Development Group
CDMA	Code Division Multiple Access
CDOS	Customer Data Operations System
CDR	Critical Design Review
CDS	Control Data Subsystem
CE	Conducted Emission
CEI	Contract End-Item
CER	Cost Estimating Relationship
CFR	Code of Federal Regulations
CFS	Cambridge File Server
CG	Center of Gravity
CIE	Customer Interface Element
CIL	Critical Item List
CIU	Customer Interface Unit
CLAN	Core Local Area Network
CM	Configuration Management
CM	Center of Mass
CMDB	Configuration Management Data Base

CMG	Control Moment Gyro
CMOS	Complementary Metal-Oxide Semiconductor
CMS	Customer Mission Specialist
CMU	Carnegie-Mellon University
CO	Contracting Officer
COF	Component Origination Form
COL	Controlled Operations Library
COMM	Commercial Missions
COP	Co-orbital Platform
COPCC	Coorbit Platform Control Center
COPOCC	COP Operations Control Center
COTS	Commercial Off-the-Shelf Software
CPCI	Computer Program Configuration Item
CPU	Central Processing Unit
CQL	Channel Queue Limit
CR	Compression Ratio
CR	Change Request
CR&D	Contract Research and Development
CRC	Cyclic Redundancy Checks
CRF	Change Request Form
CRSS	Customer Requirements for Standard Services
CRT	Cathode Ray Tube
CS	Conducted Susceptibility
CSD	Contract Start Date
CSDL	Charles Stark Draper Laboratory
CSMA/CD/TS	Carrier-Sense Multiple with Access/Collision Detection and Time Slots
CSTL	Controlled System Test Library
CTA	Computer Technology Associates
CTE	Coefficient of Thermal Expansion
CUI	Common Usage Item
CVSD	Code Variable Slope Delta (Modulation)
CWG	Commonality Working Group
D&B	Docking and Berthing
DADS	Digital Audio Distribution System
DAIS	Digital Avionics Integration System
DAR	Defense Acquisition Regulation

DARPA	Defense Advanced Research Projects Agency
DB	Data Base
DBA	Data Base Administrator
DBML	Data Base Manipulation Language
DBMS	Data Base Management System
DCAS	Defense Contract Administrative Services
DCDS	Distributed Computer Design System
DCR	Data Change Request
DDBM	Distributed Data Base Management
DDC	Discipline Data Center
DDT&E	Design, Development, Testing, and Engineering
DEC	Digital Equipment Corp.
DES	Data Encryption Standard
DFD	Data Flow Diagram
DGE	Display Generation Equipment
DHC	Data Handling Center
DID	Data Item Description
DIF	Data Interchange Format
DMA	Direct Memory Access
DMS	Data Management System
DoD	Department of Defense
DOMSAT	Domestic Communications Satellite System
DOS	Distributed Operating System
DOT	Department of Transportation
DPCM	Differential Pulse Code Modulation
DPS	Data Processing System
DR	Discrepancy Report
DR	Data Requirement
DRAM	Dynamic Random-Access Memory
DRD	Design Requirement Document
DS&T	Development Simulation and Training
DSDB	Distributed System Data Base
DSDL	Data Storage Description Language
DSDS	Data System Dynamic Simulation
DSIT	Development, Simulation, Integration and Training
DSN	Deep-Space Network
DTC	Design to Cost

DTC/LCC	Design to Cost/Life Cycle Cost
DTG	Design To Grow
E/R	Entity/Relationship
EADI	Electronic Attitude Direction Indicator
ECC	Error Correction Codes
ECLSS	Environmental Control and Life-Support System
ECMA	European Computers Manufacturing Assoc.
ECP	Engineering Change Proposals
ECS	Environmental Control System
EDF	Engineering Data Function
EEE	Electrical, Electronic, and Electromechanical
EHF	Extremely High Frequency
EHSI	Electronic Horizontal Situation Indicator
EIA	Electronic Industry Association
EL	Electroluminescent
EM	Electromagnetic
EMC	Electromagnetic Compatibility
EMCFA	Electromagnetic Compatibility Frequency Analysis
EME	Earth Mean Equator
EMI	Electromagnetic Interference
EMR	Executive Management Review
EMS	Engineering Master Schedule
EMU	Extravehicular Mobility Unit
EMUDS	Extravehicular Maneuvering Unit Decontamination System
EO	Electro-optic
EOL	End of Life
EOS	Earth Observing System
EPA	Environmental Protection Agency
EPS	Electrical Power System
ERBE	Earth Radiation Budget Experiment
ERRP	Equipment Replacement and Refurbishing Plan
ESR	Engineering Support Request
ESTL	Electronic Systems Test Laboratory
EVA	Extravehicular Activity
F/T	Fault Tolerant
FACC	Ford Aerospace and Communications Corporation
FADS	Functionally Automated Database System

FAR	Federal Acquisition Regulation
FCA	Functional Configuration Audit
FCOS	Flight Computer Operating System
FCR	Flight Control Rooms
FDDI	Fiber Distributed Data Interface
FDF	Flight Dynamics Facility
FDMA	Frequency-Division Multiple Access
FEID	Flight Equipment Interface Device
FETMOS	Floating Gate Election Tunneling Metal Oxide Semiconductor
FF	Free Flier
FFT	Fast Fourier Transform
FIFO	First in First Out
FIPS	Federal Information Processing Standards
fl	foot lambert - Unit of Illumination
FM	Facility Management
FMEA	Failure Modes and Effects Analysis
FMECA	Failure Mode Effects and Criticality Analysis
FO	Fiber-Optics
FO/FS/R	Fail-Operational/Fail Safe/Restorable
FOC	Fiber-Optic Cable
FODB	Fiber-Optic Data Bus
FODS	Fiber Optic Demonstration System
FPR	Federal Procurement Regulation
FQR	Formal Qualification Review
FSD	Full-Scale Development
FSE	Flight Support Equipment
FSED	Full Scale Engineering Development
FSIM	Functional Simulator
FSW	Flight Software
FTA	Fault Tree Analysis
FTMP	Fault Tolerant Multi-Processor
FTSC	Fault Tolerant Space Computer
GaAs	Gallium Arsenide
GaAsP	Gallium Arsenic Phosphorus
GaInP	Gallium Indium Phosphorus
GaP	Gallium Phosphorous
GAPP	Geometric Arithmetic Parallel Processor

Gbps	Gigabits Per Second
GBSS	Ground Based Support System
GEO	Geosynchronous Earth Orbit
GEP	Gas Election Phosphor
GFC	Ground Forward Commands
GFE	Government-Furnished Equipment
GFP	Government-Furnished Property
GFY	Government Fiscal Year
GIDEP	Government/Industry Data Exchange Program
GMM	Geometric Math Model
GMS	Geostationary Meteorological Satellite
GMT	Greenwich Mean Time
GMW	Generic Maintenance Work Station
GN&C	Guidance, Navigation, and Control
GPC	General-Purpose Computer
GPP	General-Purpose Processor
GPS	Global Positioning System
GRO	Gamma Ray Observatory
GSC	Ground Service Center
GSE	ground Support Equipment
GSFC	(Robert H.) Goddard Space Flight Center
GTOSS	Generalized Tethered Object System Simulation
H/W	Hardware
HAL	High-Order Algorithmic Language
HDDR	Help Desk Discrepancy Report
HDDR	High Density Digital Recording
HEP	Heterogeneous Element Processor
HFE	Human Factors Engineering
HIPO	Hierarchical Input Process Output
HIRIS	High Resolution Imaging Spectrometer
HM1	Habitation Module One
HM	Habitation Module
HOL	High Order Language
HOS	High Order Systems
HPP	High Performance Processors
HRIS	High Resolution Imaging Spectrometer
I	Interactive

I/F	Interface
I/O	Input/Output
IBM	IBM Corporation
IC	Intercomputer
ICAM	Integrated Computer-Aided Manufacturing
ICB	Internal Contractor Board
ICD	Interface Control Document
ICOT	Institute (for new generation) Computer Technology
ICS	Interpretive Computer Simulation
ID	Interface Diagram
ID	Identification
IDM	Intelligent Database Machine
IDMS	Information and Data Management System
IEEE	Institute of Electrical and Electronic Engineers
IEMU	Integrated Extravehicular Mobility Unit
IF	Intermediate Frequency
IFIPS	International Federation of Industrial Processes Society
ILD	Injector Laser Diode
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IOC	Initial Operating Capability
IOP	Input/Output Processor
IPCF	Interprocess Communications Facility
IPC	Interprocesses Communication
IPL	Initial Program Load
IPR	Internal Problem Report
IPS	Instrument Pointing System
IR	Infrared
IR&D	Independent Research and Development
IRN	Interface Revision Notices
ISA	Inertial Sensor Assembly
ISA	Instruction Set Architecture
ISDN	Integration Services Digital Network
ISO	International Standards Organization
ITAC-O	Integration Trades and Analysis-Cycle O
ITT	International Telegraph and Telephone
IV&V	Independent Validation and Verification

IVA	Intravehicular Activity
IWS	Intelligent Work Station
JPL	Jet Propulsion Laboratory
JSC	(Lyndon B.) Johnson Space Center
KAPSE	Kernal APSE
KEE	Knowledge Engineering Environment
KIPS	Knowledge Information Processing System
KOPS	Thousands of Operations Per Second
KSA	Ku-band, Single Access
KSC	(John F.) Kennedy Space Center
Kbps	Kilobits per second
Kipc	Thousand instructions per cycle
LAN	Local-Area Network
LaRC	Langley Research Center
LCC	Life-Cycle Cost
LCD	Liquid Crystal Display
LDEF	Long-Duration Exposure Facility
LDR	Large Deployable Reflector
LED	Light-Emitting Diode
LEO	Low Earth Orbit
LeRC	Lewis Research Center
LIDAR	Laser-Instrument Distance and Range
LIFO	Last In First Out
LIPS	Logical Inferences Per Second
LISP	List Processor
LisP	List Processor
LLC	Logical Link Control
LMI	LISP Machine Inc.
LN_2	Liquid Nitrogen
LNA	Low-noise Amplifier
LOE	Level of Effort
LOE	Low-earth Orbit Environments
LOS	Loss of Signal
LPC	Linear Predictive Coding
LPS	Launch Processing System
LRU	Line-Replaceable unit
LSA	Logistic Support Analysis

LSAR	Logistic Support Analysis Report
LSE	Language Sensity Editors
LSI	Large-scale Integration
LTV	LTV Aerospace and Defense Company, Vought Missiles Advanced Programs Division
LZPF	Level 0 Processing Facility
M	Manual
µP	Microprocessor
MA	Multiple Access
MA	Managing Activity
MAPSE	Minimum APSE
Mbps	Million Bits Per Second
MBPS	Million Bits Per Second
MCAIR	McDonnell Aircraft Company
MCC	Mission Control Center
MCC	Microelectronics and Computer Technology Corp.
MCDS	Management Communications and Data System
MCM	Military Computer Modules
MCNIU	Multi-compatible Network Interface Unit
MDAC-HB	McDonnell Douglas Astronautics Company-Huntington Beach
MDAC-STL	McDonnell Douglas Astronautics Company-St. Louis
MDB	Master Data Base
MDC	McDonnell Douglas Corporation
MDMC	McDonnell Douglas Microelectronics Center
MDRL	McDonnell Douglas Research Laboratory
MFLOP	Million Floating Point Operations
MHz	Million Hertz
MIMO	Multiple-Input Multiple-Output
MIPS	Million (machine) Instructions Per Second
MIT	Massachusetts Institute of Technology
MITT	Ministry of International Trade and Industry
MLA	Multispectral Linear Array
MMI	Man Machine Interface
MMPF	Microgravity and Materials Process Facility
MMS	Module Management System
MMS	Momentum Management System
MMU	Mass Memory Unit

MMU	Manned Maneuvering Unit
MNOS	Metal-Nitride Oxide Semiconductor
MOC	Mission Operations Center
MOI	Moment of Inertia
MOL	Manned Orbiting Laboratory
MOS	Metal Oxide Semiconductor
MPAC	Multipurpose Application Console
MPS	Materials, Processing in Space
MPSR	Multi-purpose Support Rooms
MRMS	Mobile Remote Manipulator System
MRWG	Mission Requirements Working Group
MSFC	(George C.) Marshall Space Flight Center
MSI	Medium-Scale Integration
MSS	Multispectral Scanner
MTA	Man-Tended Approach
MTBF	Mean Time Between Failures
MTTR	Mean Time to Repair
MTU	Master Timing Unit
NASA	National Aeronautics and Space Administration
NASCOM	NASA Communications Network
NASPR	NASA Procurement Regulation
NBO	NASA Baseline
NBS	National Bureau of Standards
NCC	Network Control Center
NFSD	NASA FAR Supplement Directive
NGT	NASA Ground Terminals
NHB	NASA Handbook
NISDN	NASA Integrated System Data Network
NIU	Network Interface Unit
NL	National Language
NLPQ	National Language for Queuing Simulation
NMI	NASA Management Instruction
NMOS	N-Channel Metal-Oxide Semiconductor
NMR	N-Modular Redundant
NOS	Network Operating System
NS	Nassi-Schneidermann
NSA	National Security Administration

NSF	National Science Foundation
NSTS	National Space Transportation System
NTDS	Navy Tactical Data System
NTE	Not To Exceed
NTRL	NASA Technology Readiness Level
NTSC	National Television Standards Committee
Nd:YAG	Neodynium Yttrium Aluminum Garnet (laser type)
O&M	Operations and Maintenance
O/B	Onboard
OASCB	Orbiter Avionics Software Control Board
OCN	Operations and Control Network, Operational Control Networks
ODB	Operational Data Base
ODBMS	Onboard Data Base Management System
OEL	Operating Events List
OES	Operating Events Schedule
OID	Operations Instrumentation Data
OLTP	On Line Transaction Processing
OMCC	Operations Management and Control Center
OMV	Orbital Maneuvering Vehicle
ONR	Office of Naval Research
ORU	Orbital Replacement Unit
OS	Operating System
OSE	Orbit Support Equipment
OSI	Open Systems Interconnect
OSM	Orbital Service Module
OSSA	Office of Space Science and Applications
OSTA	Office of Space and Terrestrial Application
OSTDS	Office of Space Tracking and Data Systems
OTV	Orbital Transfer Vehicle
P&SA	Payload and Servicing Accommodations
P/L	Payload
PA	Product Assurance
PAM	Payload Assist Module
PASS	Primary Avionics Shuttle Software
PBX	Private Branch Exchange
PC	Personal Computer
PCA	Physical Configuration Audit

PCA	Program Change Authorization
PCM	Pulse Code Modulation
PCR	Program Change Request
PDP	Plazma Display Panel
PDR	Preliminary Design Review
PDRD	Program Definition and Requirements Document
PDRSS	Payload Deployment and Retrieval System Simulation
PILS	Payload Integration Library System
PIN	Personal Identification Number
PLA	Programmable Logic Array
PLAN	Payload Local Area Network
PLSS	Payload Support Structure
PMAD	Power Management and Distribution
PMC	Permanently Manned Configuration
PN	Pseudonoise
POCC	Payload Operations Control Center
POP	Polar Orbiter Platform
POPCC	Polar Orbit Platform Control Center
POPOCC	POP Operations Control Center
PRISM	Prototype Inference System
PSA	Problem Statement Analyzer
PSA	Preliminary Safety Analysis
PSCN	Program Support Communications Network
PSL	Problem Statement Language
PTR	Problem Trouble Report
QA	Quality Assurance
R	Restricted
R&D	Research and Development
R&QA	Reliability and Quality Assurance
R/M/A	Reliability/Maintainability/Availability
R/T	Real Time
RAD	Unit of Radiation
RAM	Random Access Memory
RAP	Relational Associative Processor
RC	Ring Concentrator
RCA	RCA Corporation
RCS	Reaction Control System

RDB	relational Data Base
RDC	Regional Data Center
REM	Roentgen Equivalent (man)
RF	Radio Frequency
RFC	Regenerative Fuel Cell
RFI	Radio Frequency Interference
RFP	Request for Proposal
RGB	Red-Green-Blue
RID	Review Item Disposition
RID	Revision Item Description
RISC	Reduced Instruction Set Computer
RMS	Remote Manipulator System
RMSE	Root Mean Square Error
RNET	Reconfiguration Network
ROM	Read Only Memory
ROTV	Reuseable Orbit Transfer Vehicle
RPMS	Resource Planning and Management System
RS	Reed-Solomon
RSA	Rivest, Skamir and Adleman (encryption method)
RTX	Real Time Execution
S&E	Sensor and Effector
S/C	Spacecraft
S/W	Software
SA	Single Access
SA	Structured Analysis
SAAX	Science and Technology Mission
SAE	Society of Automotive Engineers
SAIL	Shuttle Avionics Integration Laboratory
SAIS	Science and Applications Information System
SAR	Synthetic Aperture Radar
SAS	Software Approval Sheet
SASE	Specific Application Service Elements
SATS	Station Accommodations Test Set
SBC	Single Board Computer
SC	Simulation Center
SCR	Software Change Request
SCR	Solar Cosmic Ray

SCS	Standard Customer Services
SDC	Systems Development Corporation
SDP	Subsystem Data Processor
SDR	System Design Review
SDTN	Space and Data Tracking Network
SE&I	Systems Engineering and Integration
SEI	Software Engineering Institute
SESAC	Space and Earth Scientific Advisory Committee
SESR	Sustaining Engineering System Improvement Request
SESS	Software Engineering Standard Subcommittee
SEU	Single Event Upset
SFDU	Standard Format Data Unit
SI	International System of Units
SIB	Simulation Interface Buffer
SIFT	Software Implemented Fault Tolerance
SIMP	Single Instruction Multi-Processor
SIRTF	Shuttle Infrared Telescope Facility
SLOC	Source Lines of Code
SMC	Standards Management Committee
SMT	Station Management
SNA	System Network Architecture
SNOS	Silicon Nitride Oxide Semiconductor
SNR	Signal to Noise Ratio
SOA	State Of Art
SOPC	Shuttle Operations and Planning Complex
SOS	Silicon On Saphire
SOW	Statement of Work
SPC	Stored Payload Commands
SPF	Software Production Facility
SPF	Single-Point Failure
SPR	Spacelab Problem Reports
SPR	Software Problem Report
SQA	Software Quality Assurance
SQAM	Software Quality Assessment and Measurement
SQL/DS	SEQUEL Data System
SRA	Support Requirements Analysis
SRAM	Static Random Access Memory

SRB	Software Review Board
SRC	Specimen Research Centrifuge
SREM	Software Requirements Engineering Methodology
SRI	Stanford Research Institute
SRM&QA	Safety, Reliability, Maintainability, and Quality Assurance
SRMS	Shuttle Remote Manipulator System
SRR	System Requirements Review
SS	Space Station
SSA	Structural Systems Analysis
SSA	S-band Single Access
SSCB	Space Station Control Board
SSCC	Station Station Communication Center
SSCR	Support Software Change Request
SSCS	Space Station communication system
SSCTS	Space Station communications and tracking system
SSDMS	Space Station data management system
SSDR	Support Software Discrepancy Report
SSDS	Space Station data system
SSE	Software Support Environment
SSEF	Software Support Environment Facility
SSIS	Space Station Information System
SSME	Space Shuttle Main Engine
SSO	Source Selection Official
SSOCC	Space Station Operations Control System
SSOCC	Space Station Operations Control Center
SSOL	Space Station Operation Language
SSON	Spacelab Software Operational Notes
SSOS	Space Station Operating System
SSP	Space Station Program
SSPE	Space Station Program Element
SSPO	Space Station Program Office
SSSC	Space Station Standard Computer
SSST	Space Station System Trainer
STAR	Self Test and Recovery (repair)
STARS	Software Technology for Adaptable and Reliable Software
STDN	Standard Number
STI	Standard Technical Institute

STO	Solar Terrestrial Observatory
STS	Space Transportation System
SUSS	Shuttle Upper Stage Systems
SYSREM	System Requirements Engineering Methodology
Si	Silicon
SubACS	Submarine Advanced Combat System
TAI	International Atomic Time
TBD	To Be Determined
TBU	Telemetry Buffer Unit
TC	Telecommand
TCP	Transmissions Control Protocols
TCS	Thermal Control System
TDASS	Tracking and Data Acquisition Satellite System
TDM	Technology Development Mission
TDMA	Time-Division Multiple Access
TDRS	Tracking and Data Relay Satellite
TDRSS	Tracking and Data Relay Satellite System
TFEL	Thin Film Electrolumenescent
THURIS	The Human Role in Space (study)
TI	Texas Instruments
TM	Technical Manual
TM	Thematic Mapper
TMDE	Test, Measurement, and Diagnostic Equipment
TMIS	Technical and Management Information System
TMP	Triple Multi-Processor
TMR	Triple Modular Redundancy
TMS	Thermal Management System
TPWG	Test Planning Working Group
TR	Technical Requirement
TRAC	Texas Reconfigurable Array Computer
TRIC	Transition Radiation and Ionization Calorimeter
TSC	Trade Study Control
TSIP	Technical Study Implementation Plan
TSP	Twisted Shielded Pair
TSS	Tethered Satellite System
TT&C	Telemetry, Tracking, and Communications
TTC	Telemetry Traffic Control

TTR	Timed Token Ring
TWT	Traveling-Wave Tube
U	Non-restrictive
UCC	Uniform Commercial Code
UDRE	User Design Review and Exercise
UIL	User Interface Language
UON	Unique Object Names
UPS	Uninterrupted Power Source
URN	Unique Record Name
UTBUN	Unique Telemetry Buffer Unit Name
UTC	Universal Coordinated Time
V&V	Validation and Verification
VAFB	Vandenberg Air Force Base
VAX	Virtual Address Exchange
VHSIC	Very High-Speed Integrated Circuit
VLSI	Very Large-Scale Integration
VLSIC	Very Large-Scale Integrated Circuit
VV&T	Validation, Verification and Testing
WAN	Wide Area Network
WBS	Work Breakdown Structure
WBSP	Wideband Signal Processor
WDM	Wavelength Division Multiplexing
WP	Work Package
WRO	Work Release Order
WS	Workstation
WSGT	White Sands Ground Terminal
WTR	Western Test Range
XDFS	XEROX Distributed File System
YAPS	Yet Another Production System
ZOE	Zone Of Exclusion
ZONC	Zone Of Non-Contact
ZnS	Zinc Sulfide

Volume II
TASK 2 - OPTIONS DEVELOPMENT
2.0 DESIGN OPTIONS

SUMMARY

This volume contains the options development for the Design Options Category. The specific design options and their respective volume II section numbers are as follows:

- 2.1 Software
  - 2.1.1 Data Base Management
  - 2.1.2 Resource Management
  - 2.1.3 Distributed Operating System
- 2.2 System Architecture
  - 2.2.1 Fault Tolerance
  - 2.2.2 Autonomy/Automation
  - 2.2.3 System Growth
  - * 2.2.4 Deleted
  - 2.2.5 System Interfaces
  - 2.2.5.1 SSDS/Payload
  - * 2.2.5.2 Deleted
  - 2.2.5.3 Man/Machine (workstations)
  - * 2.2.5.4 Deleted
- 2.3 System Security/Privacy
- 2.4 Time Management
- 2.5 Communications
  - 2.5.1 Space Communications
  - 2.5.2 Wide Area Communications
  - 2.5.3 Local Area Networking
- 2.6 Network Performance Assessment

For the options development general approach and methodology the reader is referred to the introductory sections of Task 2, Options Development, Volume I.

* These items have been deleted or incorporated into other sections.

2.0 DESIGN OPTIONS

2.1 SOFTWARE

PRECEDING PAGE BLANK NOT FILMED

DATA BASE MANAGEMENT (OPTIONS 2.1.1)

2.1.1.1.0 INTRODUCTION AND APPROACH

The Data Base Management Systems (DBMS's) that will be used in the Space Station Program (SSP) will probably not be designed from basic data base principles but rather will be vendor products (possibly modified for a specific application). This is necessary to realize cost effectiveness. There has also been a great deal of investment in this area in the commercial world. However,it is still necessary to understand the principles of DBMS's to select the commercial products which are best suited for the "user driven" data manipulation requirements in the SSP. The characterization of user data manipulation requirements and the characterization of the data collection method and location(s) are the primary drivers in determining the desired DBMS characteristics such as:

data structure
distribution/partitioning
replication/recovery
interface
presentations and reports

The details of various vendor options can be traded to establish the best match to the requirements. The problem boils down to understanding the features of various vendor options and understanding the diverse requirements of various SSP SSIS data handling entities. These SSIS entities are distinquishable because of unique data views and locations. Some segment(s) of the SSP DB exist(s) at each SSIS entity. The list of entities identified on the SSP is given in section 5.1. At each of these entities there will be individuals(or teams of individuals) called data base administrators (DBA's) with the responsibility to make the data base design decisions. These DBA's will have to understand the requirements of DB users and available vendor options to make decisions. This paper is written to aid the DBA's in these decisions.

The approach taken in this SSDS A/A study is to consider all SSIS DB entities and then to concentrate on the characteristics of the SSDS DB embedded within the SSIS. For the data base problem this approach translates into the following study stages:

1. Define all the SSIS DB entities using the TASK 1 functions list and the basic premise that there will be partitioning and geographic distribution of TASK 1 level0 functions.

2. Determine which of these DB's are within the SSDS (exclude TMIS segments) while still considering required connectivity and interfaces of all segments.

3. Characterize the SSDS DB entities by:
   i) data type and source
   ii) functional manipulation requirements
   iii) connectivity

   using the supporting material in Appendix B.

4. Define options for commercial products applicable to the ground segments.

5. Define options (commercial, modified commercial or "roll your own") for space segments.

The data base design process will then continue into the SSDS A/A TASK 3 and TASK 4 where trades will be performed and a design will be recommended. TASK 3 and TASK 4 will have the following stages:

6. Partition the SSDS DB's by space/ground segments

7. Partition the SSDS DB ground segments to as low a level as required to separate by utilization (development, operational, scientific/PL) and also by interest domains, data types and DBMS functional requirements (see section 1.5)

8. Trade options and define recommended SSDS DB architecture.

2.1.1.1.1 PROGRAM REQUIREMENTS FOR DBMS's

1.1.1 TRACEABILITY TO FUNCTION LIST

As part of the SSDS A/A Study Task No.1 a list of functions has been developed. In Appendix A the functions requiring DBMS support are listed. An assumption has been made that data being processed in a transport service such as a telemetry stream or communication network where short term buffering is required is not a part of the DBMS.

1.1.2 CUSTOMER REQUIREMENTS FOR STANDARD SERVICES FROM THE SSIS

Appendix B contains extracts from the SSIS/SCS document prepared at GSFC. This material defines the DB segments to support the "short and long term" storage requirements of customer data.

1.1.3 SPACE STATION REFERENCE CONFIGURATION

The Space Station Reference Configuration document suggests some characteristics for the onboard data base. The sections of this document describing the data base characteristics are included in Appendix B.

1.1.4 SS DEFINITION & PRELIMINARY DESIGN RFP

The Phase B RFP describes functional attributes of the DBMS and these sections are included in Appendix B.

1.1.5 SSIS DATABASE MANAGEMENT SERVICE

The SSIS final report (JPL D-1737) contains recommended characteristics of the DBMS services and these are included in Appendix B.

1.2 TMIS

The TMIS can be mapped to functions defined to be within the SSIS. The data base maintained by the TMIS corresponds to section 7.5 of the functions list, that is "Configuration Management". The TMIS is considered to contain the following data segments:

TMIS DATA CONTENT

LEVEL B SE&I MASTER DATA BASE(MDB)

- LEVEL A SPEC
- SSIS CONFIG(CONNECTIVITY,ICD's)
- REF CONFIG(DRAWINGS,TEXT)
- WP ICD's

LEVEL B SE&I ENGINEERING MASTER SCHEDULE(EMS)

- SE&I SCHEDULES
- S/W SCHEDULES
- HARDWARE SCHEDULES

LEVEL C SE&I DATA BASE

- HARDWARE SPEC's
- SUBSYSTEM ICD's
- S/W REQUIREMENTS
- SSE REQUIREMENTS

The functional description of TMIS provided in the Phase B RFP is included in Appendix B. The TMIS functions equate closely to the "development" portion of the major partitioning described in Table 5.1.3. The TMIS DBMS is not considered to be a part of SSDS but there there may be some data exchange required from the SSDS DB segments to support TMIS. It is not known at this time what that data might be.

1.3 SCOPE OF DATA BASE MANAGEMENT

A basic issue that needs to be addressed concerning data created as a part of the SSP is " should this data be included under a DBMS that allows the data to be shared or should it remain under control of the owner in some simple file handler"?. A lot of data will be created which is of no interest to anyone but the originator. There may also be data only of interest to a small group. Should that data be managed by this group or a central authority? An example might be a special analytical tool developed by a contractor. These issues must be addressed to establish the scope of each DB segment and the partitioning of authority for administration of each segment. To establish reachability of data the need for remote access must be demonstated. In section 1.2 the segments of TMIS where described along with the contents of each segment. This list gives the characteristics of data that will be shared and should reside in a DB. In this section the DB's other than TMIS are characterized and the data described is considered shared data which should reside in the data base.

The need for data bases other than TMIS is clear from the Task 1 requirements (Appendix A). The following preliminary partitioning along functional lines is proposed:

DATA BASES OTHER THAN TMIS

SSE DB

- SOFTWARE
- MODELS
- TEST SCRIPTS
- RESOURCE SCHEDULING

TRAINING DB

- PROCEDURES
- SCHEDULES

INTEGRATION SITE

- SOFTWARE
- INTEGRATION SCHEDULES
- PROCEDURES
- TEST SCRIPTS

SSCC

- SPACE STATION STATUS
- MISSION SEQUENCING
- COMMAND PROCEDURES

POCC DB

- PLATFORM/PL ENGINEERING DATA
- EXPERIMENT DATA

REGIONAL DATA CENTERS/DISCIPLINE DATA CENTERS

- SPACE STATION ANCILLARY DATA ARCHIVE
- PLATFORM ANCILLARY DATA ARCHIVE
- FF ANCILLARY DATA ARCHIVE

SPACE STATION O/B DATA BASE

- DOCUMENT MANAGEMENT
    - MANUALS(PROCEDURES)
    - DAILY SCHEDULES
    - DIAGNOSTIC SUPPORT(SCHEMATICS)
- SOFTWARE
- CHECKPOINTS
- SUBSYSTEM TREND DATA
- REAL-TIME DATA
- BUFFERED DATA(RECORDERS)

CO-ORBITING/POLAR-ORBITING PLATFORM O/B DATA BASE

- ENGINEERING DATA
- SCIENTIFIC DATA

DATA HANDLING FACILITY

- LEVELO DATA
- SHORT TERM ARCHIVE
- LONG TERM ARCHIVE

1.4 DATA BASE ADMINISTRATION

There will be some one identifiable person (or team of people) who has the central responsibility for the "operational data" (i.e. the released data viewed by DB users). This person is the data base administrator (DBA). The centralized control of the operational data performed by the DBA provides the capability to:

Reduce redundancy
Avoid inconsistency
Share data with new applications

Enforce standards

Apply security restrictions

Maintain integrity

Structure for performance for most used applications

The role of the DBA is important and a high degree of technical expertise is required. The DBA responsibilities include:

Deciding information content of DB

Deciding storage structure and access strategy

Coordinate with users

Define authorization checks and validation procedures

Define strategy for backup and recovery

Monitor performance and respond to requirements changes

The DBA must have the techical expertise to understand the applications of the data being stored and manipulated and also be a DB expert. The DBA is supported by a number of utility programs which are an essential part of a practical DBMS. One of the most important DBA utilities is the "data dictionary" which is effectively a DB containing descriptions of the DB content.

For the SSP there will be many DBA's, each with responsibility for one of the DB's described in sections 1.2 and 1.3 or some segment of a DB. The assignment of these individuals or teams should be done in parallel with major DB partitioning and should be done early in the program before DB's start emerging with no visible control mechanism (i.e. DBA). This mainly concerns TMIS where system requirements are emerging during phase B.

1.5 DATA CHARACTERISTICS

The characterization of data within a data base will aid in determining the data base system features required to manipulate this data. The table below contains the three basic data types, their attributes and the functions needed from the DBMS. Each DB entity contains segments. Each segment contains just one of these data types.

TABLE 1.5.1 DATA CHARACTERIZATION

TYPE	ATTRIBUTES	DBMS FUNCTIONS REQUIRED
SAMPLED DATA	o NUMERICAL DATA RECORDS FROM EXP OR OPERATIONAL SENSORS o ANCILLARY DATA RECORDS	o STORE o RETRIEVE o COPY o ARCHIVE o CREATE FILE o KEEP DIRECTORY
TEXT	o ALPHA NUMERIC CHARACTERS ENTERED AND STORED BY LINE AND PAGE AND DOCUMENT	o WORD PROCESSING o STORE o RETRIEVE o COPY o PRESENTATION o CREATE FILE o KEEP DIRECTORY
ASSOCIABLE	o TABLES OF DATA IN WHICH LOGICAL INFERENCES CAN BE MADE FROM CORRELATION	o AD HOC QUERY o MANIPULATE - JOIN - PROJECT - RESTRICT o PRESENTATION

Data can also be broadly categorized by major phase or area of the program:

i) Development (dev schedules, dev status, analysis, specs, ICD's)
ii) Operational (activity sch, vehicle status)
iii) Scientific (experiment data)

Table 5.1.3 illustrates a partitioning along these lines. This subdivision will aid in determining the range of visibility needed within each data base and thus network connectivity. Some data can logically fall into multiple categories in which case this categorization may not help partitioning so much. As an example, operational data contains the ancillary data and will be of interest to the scientific community. The ancillary data will also be used for vehicle monitoring. There is also a good deal of overlap within the development and operational areas. This grouping can still be used as a high level segmentation to minimize constaints on DB selection caused by coupling diverse requirements.

2.1.1.2. TECHNICAL ATTRIBUTES OF OPTIONS

2.1 DATA MODEL (STRUCTURE) (reference 18)

The data model is the logical method used to organize and retrieve data. It has two aspects:

i) How the data is organized and structured
ii) How the data is accessed

There are four options:

i) Relational
ii) Hierarchical
iii) Network
iv) Inverted

The selection of a data model depends on the nature of the application. Jeff Stamen of Management Decision Systems suggests the following generic selection criteria:

Planning and analysis ————| Decision Support Systems
(graphics,word processing,
electronic mail,etc.)

Management Control ——————| Relational

Operational ————————| Structured (hierarchical, network,
inverted)

2.1.1 RELATIONAL

The relational model introduced by E.F. Codd (reference 17) in 1970 has demonstrated many features desireable in a DBMS: flexibility for growth with minimal impact, ad-hoc query. A tutorial on the distributed relational DB has been prepared and included as Appendix C. This material is from reference 1.

2.1.2 HIERARCHICAL

This model is the original model for data and is widely used. It presents the fastest access of any model along the primary path (i.e., a direct search to a piece of data from the root). It responds best to data searchs desribed as "many to one" but doesn't handle "many to many" well. It can use a lot of redundant data storage. It is relatively inflexible requiring much preplanning for performance. The strengths of this model are for large data bases with many users queries being serviced. For DB's where the data is either "sampled data" or "text" (as described in section 1.5) this model presents the best performance. Much of the "operational" and "scientific" data will fall in these categories.

2.1.3 NETWORK

The network solution for general usage is better than the hierarchical although not as fast on the primary path. It is much better for multiple search strategies. The network model can represent the hierarchical "many to one" search as a subset. It can represent any relationship easily and efficiently. However, it is difficult to add, change or delete relationships. It is also difficult to add, change or delete files.

2.1.4 INVERTED

An inverted file is one in which access to record occurrences can be gained by presentation of one or more data item values (keys). For example, a bibliographic file might be inverted on three keys: title, author, and subject. A file is "totally inverted" if access can be gained through any data item value. Otherwise, it is "partially inverted". The inverted model is as flexible for growth as the relational model. Some implementations are more mature than the relational model(i.e., in the areas of: on-line DB, query, large DB's. The inverted model is applied especially well to multi-key retrieval. If the DB is highly dynamic the inverted model can result in high overhead.

2.2 DISTRIBUTION AND PARTITIONING

Data bases may be "fully partitioned" or "fully replicated" or something inbetween. In the fully partitioned case each database segment is stored in exactly one place. In the fully replicated case each database segment is stored in all locations. Replication is not desirable if updates are common because of several problems: difficulty in keeping copies identical and network delays. However, replication is used to improve performance and as a recovery technique. These conflicting issues must be traded in DB design. Distributed data base systems are an important application of computer networking. Distributed data base systems infers that there is some geographical distribution of the data base. This distribution can present the DB designer with several advantages:

i) increased reliability because redundancy is build in to tolerate hardware failures

ii) improved transaction response because we are dealing with multiple machines

iii) incremental growth options on a collection of smaller systems

2.2.1 CONCURRENCY CONTROL

There are three basic strategies for ensuring the consistency of a DB.

i) The first strategy is based on two-phase locking and requires that a transaction acquire an exclusive lock on each data object that it wishes to read before releasing any lock. The locks are usually acquired on demand during run time. In order to facilitate easy recovery the locks are usually held by the transaction until commit.

ii) The second strategy assigns each transaction a time-stamp, which is unique throughout the system. Transactions are required to execute in the order of their time-stamps.

iii) The third stategy, which is usually called optimistic, allows each transaction to execute freely and during the transaction execution and/or at the end of it the objects read and updated by the transaction are reviewed to ensure that the transaction viewed the data base in a consistent state. If so, the transaction commits. If not, the transaction is aborted and restarted.

In reference 22 there is a discussion of these concurrency techniques. This paper addresses the problem of maintaining consistency in distributed data bases. An algorithm is proposed which operates as an optimistic concurrency control until the rate of transaction contentions reaches a specified degree. The algorithm then switches to a pessimistic mode similar to a two-phase locking. The main arguments in favor of the optimistic concurrency control are:

i) it provides higher degree of concurrency and therefore smaller delays

ii) the overhead associated with non-conflicting transaction synchronization is low although the overhead for conflicting transactions can be high

The optimistic algorithm is reported to perform better than the other control mechanisms for the following reasons:

i) It produces minimum synchronization overhead for non-conflicting transactions.

ii) The proposed algorithm achieves some increase in concurrency over two-phase commit by allowing a transaction to access results generated by other transactions which are not yet committed.

iii) The proposed algorithm adapts timewise and spacewise and thus it can switch between pessimistic (which uses long term locks) and optimistic (which uses short term locks) modes at any time and over any data object.

Concurrency control in distributed databases is also discussed in Appendix C where several references are sited. Much research and many new products are being announced in the area of distributed relational data bases and these need to be evaluated by the DB designer. Reference 19 suggests that there are still some technological hurdles to overcome before general application is achieved.

2.3 PERFORMANCE

To a data base user the main metrics which can be applied to any DB are (in order of importance):

i) availability

ii) ease of use

- change
- ad hoc query

iii) response time

- query
- update

These are all reasonable measures for evaluating a DBMS and can be used by the designer to evaluate options after setting up a set of "bench mark" applications typical of the DB users. To insure good performance for the user within cost the DB designer must also be provided with measures on such characteristics as:

i) reliability

ii) throughput

iii) storage efficiency

iv) flexibility for growth

v) cost

The best way to compare products is to present the same "problems" to to them and measure the metrics described above. Some of the measures must be provided by the vendor (i.e., reliability, cost).

2.4 SECURITY

The Data Base Administrator (DBA) ensures that the only means of access to the DB is through proper channels. The DBA defines the authorization checks to be carried out whenever access to sensitive data is attempted. Different checks can be established for each type of access (retrieve, modify, delete, etc.) and access authorization can be to some DB segment level. Security is also augmented by the Operating System (OS) and the network manager. Communication with the DB can be checked and validated at the network level if remote access is allowed.

2.5 NETWORK COMMUNICATION

The distributed DB must communicate with geographically separated segments which is usually accomplished by a "wide area network" (WAN). Reference 1 discusses this communication in the light of the ISO/OSI reference model. The distributed data base can be considered to be in the "application layer" of this model. In reference 1 page 440 Tanenbaum states, "Furthermore, there are almost no national or international standard protocols for layer 7". Figure 2.5.1 shows that a distributed data base is coupled to the features of the communication software supporting a network interface. Although this should be transparent to the user, it is not transparent to the system design and communication between distributed data bases with different partitions managed by heterogeneous DBMS's requires communication gateways and DB translation before data can be exchanged.

There are many vendors who provide the communication software to interface with their own products (remote terminals and remote computers). These products are all claimed to meet the ISO/OSI reference model but these products will not communicate at the layer 7 level.

Homogeneous distributed DBMS's must be considered as a option for some of the major segments of the SSP DB (i.e., development, operational, scientific/PL) to minimize complexity of data exchange.

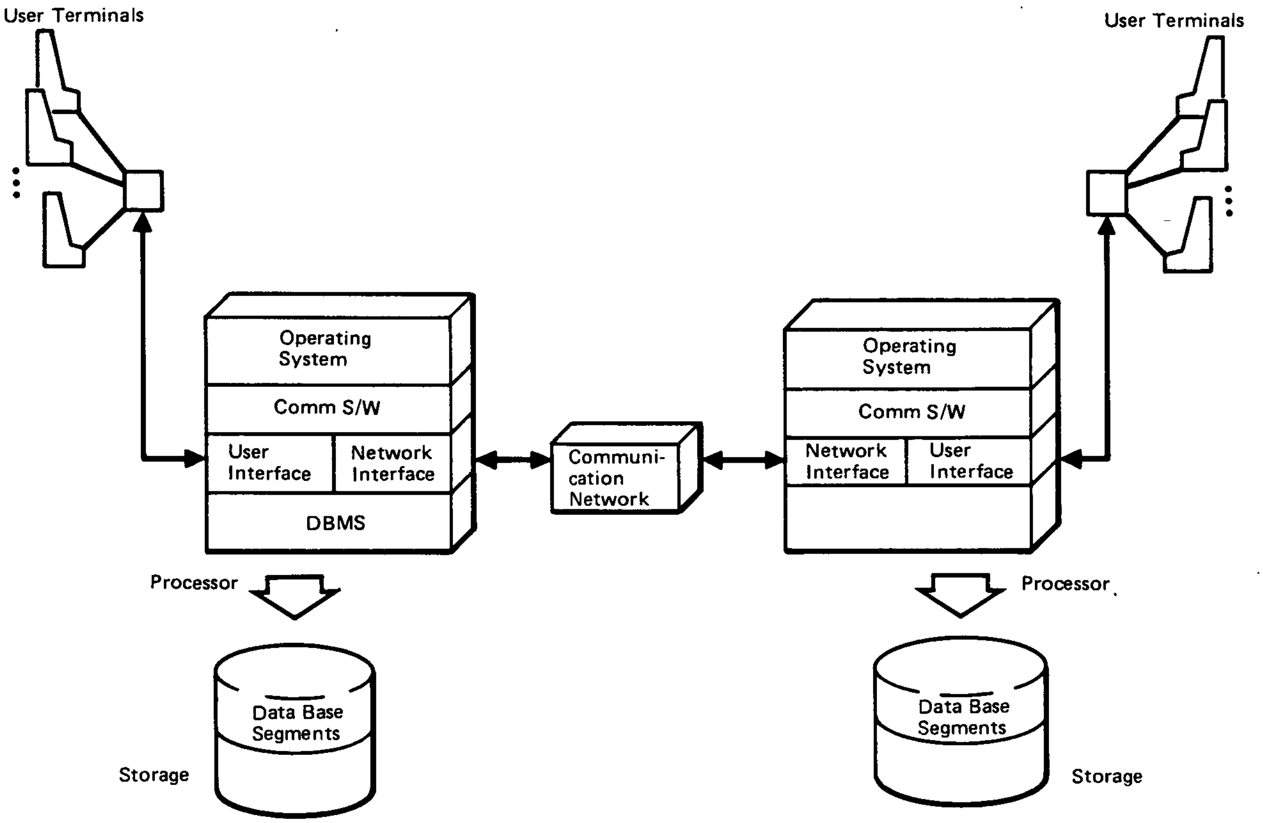


Figure 2.5.1. Distributed Data Base Management Systems

Using a WAN to exchange data between DB's requires that the addresses of these DB's be known to the network and that these addresses be unique. Reference 23 treats this topic and suggests an algorithm for dynamic address assignment. The treatment is for network stations but this could be extented to DBMS's. The delivery of data to remote sites on a public network is an option that must be considered. Currently, point-to-point arrangements are made through a service called NASCOM. This usually entails communication on leased carriers (physical layer) and the protocol at higher ISO OSI must be negotiated by the two communicating applications. Another option is to allow addressing data in a generalized end-to-end protocol which allows delivery of customer data generated in space to the customer DB using standard formats (as suggested in reference 5). The development of the Program Support Communications Network (PSCN) is a possible physical layer to start developing this generalized delivery service.

There are currently supported networks which allow remote terminal access to networks (i.e., ARPANET TELNET). These higher level protocols should be considered for remote file access. (see references 24, 25, 26).

2.6 DATA ACCESS AND USER INTERFACE LANGUAGES

A single high-level language should be considered as a query language. This language would have the capability for: definition, manipulation, and control. The features that should be looked for in a interface language are as follows:

i) Simplicity
- ease and conciseness
- elimination of lower language features

ii) Completeness
- for queries, user never has to use loops or branchs

iii) Non-procedural
- specify "what" is wanted, not procedure to get data
- high level statements of intent
- capture intent and apply search optimization

iv) Ease of extension

- provision of built in functions
- permit user defined built in functions

v) Support Higher-level Languages

- casual user natural language

In reference 26 a unified language supporting interface to all three of the commonly used data structures (i.e. relational, hierarchical, network) is proposed. This is an option to consider. In addition, programming language support must be considered. The options here are as numerous as programming languages themselves but for sure the following options need special consideration:

i) FORTRAN (for scientific and operational DB's)
ii) COBOL (for management control DB's)
iii) Assembler (for developing efficient frequently used routines)

There is a possibility to use a natural language such as English as a query language. An IBM product is available called INTELLECT. This is a natural English language query program with the following attributes: analyzes grammar, uses application-specific dictionaries, interfaces to SQL/DS data bases (see vendor products for description of SQL/DS). In reference 20 the product INTELLECT and its capabilities are briefly described. The author is with Artifical Intelligence Corporation the developer of INTELLECT. In this paper the author describes some interesting problems encountered when installing INTELLECT and has some recommendations for future research.

2.6.1 AUTOMATIC COLLECTION OF OPERATIONAL DATA

The acquisition of onboard operational data using a DMS service available to the core subsystems and users is described in reference 15. This DMS service acquires data on a periodic or aperiodic basis and also controls the interface between the operational DB and the telemetry service. This generalized service should be considered as an option for data collection into DB files since the management and location of these files must be known to the DMS function supporting the data acquisition for telemetry.

2.6.2 REPORTS AND PRESENTATIONS

The software programs to generate reports and format presentations are supplied as built-in functions to the interface language. The user should not have to resort to building special formats.

2.7 MIGRATION TO ARCHIVE

Data collected on a periodic basis will require rapid access for some applications and therefore will have to be accessible by the host computer in an "on-line" memory. Following retention for some period this data will then migrate to an archive (i.e., massive data storage with relatively slow accessibility). This migration may be done automatically by a DBMS function. The need for archive management appears in the requirements of reference 21. The process of archive management is described in reference 11. In this paper archiving strategies are discussed including such factors as:

i) archiving files that have been idle for longest time
ii) large files archived before small files
iii) periodic archiving for reliability of system
iv) archiving during low usage times

This is a general treatment but the features to look for in DB products are clearly defined along with a suggested system design. A review of this paper will aid in evaluating vendor options.

2.8 RECOVERY AND AVAILABILITY

Availability refers to the probability that the DBMS will be operating at any given time and therefore available for use. This concept is of course very closely tied to the reliability and fault tolerance of the DBMS. Recovery is the technique built into the system to recover and continue operation following failures.

There are two types of failures that the DBMS must recover from:

i) system failures
ii) media failures

The usual option offered for system failures is to provide automatic system recovery. This is done with back-up computers (redundancy) and supported by operating system functions. High reliability computers still do not preclude the need for redundancy. System crash recovery is closely coupled with concurrency control since multiple hosts must coordinate the recovery. An example of an integrated concurrency and recovery algorithm is given in reference 27 and summarized in reference 1. Many algorithms for maintaining integrity of a DB require certain actions to be "atomic", (i.e., either carried out to completion or not carried out at all). Since an atomic action may have to perform several disk accesses, there is a possibility that the system may crash part way through the atomic action, leaving the DB in an inconsistent state. In reference 28 Lampson has studied this problem and suggested an algorithm that maintains two copies of critical disk blocks in such a manner as to ensure consistency at all times. Reference 7 also discusses distributed system crash resistance and proposes a structuring to ensure consistency. The ARCHONS project at Carnegie-Mellon University is another example of research in the area of distributed system consistency (see reference 8). Reference 16 is a good survey of the concurrency control problem.

Media failures refer to the crashing of on-line memory. The option for media failures is to provide archive back-up. This is an option usually offered with vendor products.

2.1.1.3. MAJOR OPTIONS DECISIONS

3.1 DISTRIBUTION/CONNECTIVITY

One major decision will be to define to what extent each data base is reachable from various geographic and space locations. This aspect of data sharing will drive network design and universal naming conventions for the data structures (data sets or relations). The complexity of DB interfaces will also be driven by whether we have homogeneous DB's or heterogeneous DB's. Once data has been categorized as private (that is to say it is only of interest to a local individual or small group) then this data will be managed locally and the selected manager may not support communication with other managers making this data unreachable by remote interests. This presents a problem in defining early in the program all data which has a potential of being shared by decentralized users to insure the network and distributed DBMS's accommodate any future change in interest domains while still not dictating that all data be reachable from anywhere(potentially a cost driver in the communications design and constraint on using heterogeneous DB's).

3.2 ADMINISTRATION

The issue here is to insure that the authority for data base management is established early in the program and this authority is not distributed so widely that DB state becomes difficult to control. The assignment of administrators needs to be done as DB entities are defined and clear partitions are established.

There is a continuum of options here ranging over the spectrum of possible granularities given to the DB segments. Some common sense must be applied to assigning administrative authority over DB segments.

3.3 REPLICATION/RECOVERY

There is a possibility to implement a recovery scheme by augmenting a commercial product. This option can be considered if the survey of products establishes that inadequate recovery is provided or if a vendor option is selected because of superior capabilities other than recovery and augmentation is appropriate. Additional back-up by replication of the DB is an option. This may be accomplished by replication in the archive.

3.4 PARTITIONING

3.4.1 SPACE/GROUND

Partitioning of data between space and ground is a major design decision. Data storage and large DBMS software packages in space could present a higher cost for computational and storage devices because they must be space qualified. This must be traded against the bandwidth needed for space to ground transfers and queries. The response time for space queries will have to be analyzed to determine if acceptable times are realizable. For the most part this seems possible since the majority of data exchange will be non-interactive (i.e., mainly large text block transfers). The possibility of having the capability to put large DBMS in in space in the 1990's is also real (i.e., space rated memory may become less expensive) which would change the trade criteria to maintainability and demonstrating the need for space autonomous data manipulation (which may be difficult to demonstrate considering the limited crew time resource). It appears that the Space Station and other orbital elements should be treated as a place where scientific and operational data are generated but the need for manipulation (other than management for remote delivery) must be demonstrated (i.e, need for "quick look analysis"). Transformation of scientific data (from time domain to frequency domain) should be considered to be outside DBMS services.

3.4.2 GROUND SEGMENTS

The major partitioning of DB segments to ground entities (DHC, SSCC, POCC,...) presents a high level design decision that must be made early in the program to maintain stability in DB development. In section 1.3 the content of the SSP major DB segments is suggested but furthur detail of data content and exchange mechanisms must be established.

3.4.3 ANCILLARY DATA GROUPING

The performance of an O/B system which allows users an option to acquire ancillary data an append that data to experiment data will be highly dependent on the nature of the ancillary data blocks. This presents the O/B DB designer with decisions concerning ancillary file structures and granularity of ancillary data access. A common block required by most users is the vehicle state (attitude, position vector, time). Other groupings need to be established after user requirements are understood. These groupings should be such that the O/B data network is not loaded down with data transfers containing a majority of data that will be discarded. Reference 21 suggests the typical ancillary data required by customers.

3.5 ARCHIVE RESPONSIBILITY

A major DB design decision that has impact at the program level is to determine where the functional responsibility for archiving data resides. The options are some reasonable assignment of the following data groups to the major data handling centers.

DATA	CENTERS
Engineering	Data Handling
Ancillary	Space Station Control
Customer	PL Control
	Regional/Discipline

3.6 RECORDER MANAGEMENT

The option to provide bulk recording in the space segments of the DB to aid in managing the telemetry link is an area for consideration and couples tightly with the O/B DB design. The management of these recorders is another area needing consideration (DMS or C&T).

3.7 O/B DB OF SUBSYSTEM HISTORY DATA

Another decision facing the O/B DB design is to establish how much and what subsystem history data will be held O/B for O/B status support. This design decision couples with the need for O/B autonomy and the communication system capability to support interactive communication with the ground segments of the DB.

3.8 COMPATIBILITY OF DB's

The use of heterogeneous institutional facilities is a major decision. Existing data base management systems are to be provided to the Space Station Program by the Level C centers. This means dealing with heterogeneous DBMS's and the operating systems that they run under. This complicates the problem but is probably unavoidable since a variety of vendors are established at the centers and any further expansion of center capability could contain multiple vendor equipment. Compatibility with the JSC TMIS DBMS is a key driver since this DB will contain the Level B SE&I Program Master Data Base(MDB) and the Engineering Master Schedule (EMS). NASA has specified that the TMIS be IBM compatible (reference Phase B RFP C-6 3.0a). Some work is proceeding in the area of virtual terminal protocols and protocols for file transfer in heterogeneous networks (references 25, 26). Establishment of higher level protocols for resource sharing in heterogeneous systems is a possibility to ease the complexity of heterogeneous system data exchange (as suggested in reference 26).

3.9 STRUCTURE

The selection of a DB structure for each of the SSP DB's will be an important decision and require a complete understanding of the data characteristics and user intentions for data manipulation. A selection criteria suggested in reference 18 is depicted in Figure 3.9.1. Any data structure selected can be abused an result in poor performance if other factors are not considered. The organization of data within the constraints of the DBMS features can be called the DB architecture. It may turn out that this architecture is more important to performance than the DBMS data structure. The big decision that will be encountered designing the various DB'sfor the SSP (besides the selection of a DBMS) will be the characterization and desired organization (i.e., architecture) of the data within each DB segment. Data can be broadly categorized into three groups:

i) Sampled data (sequential numerical; sensor data)
ii) Text (alpha numeric; s/w prgms, presentations)
iii) Associable (tables with correlatable data; mission data)

This characterization will aid in determining the DBMS functions needed to manipulate each category of data. For category 1 a flat file server may be an adequate DB manager option Category 2 requires more DB manager services mainly related to word processing. Category 3 represents data which must be organized into records or tables so additional information can be extracted by queries which result in presentations (reports) to the user. For category 3 we must decide on the data storage structure (i.e., relational, hierarchical or network). This decision and the organization of data within that structure (i.e., architecture of DB) will determine the DB performance (throughput and response).

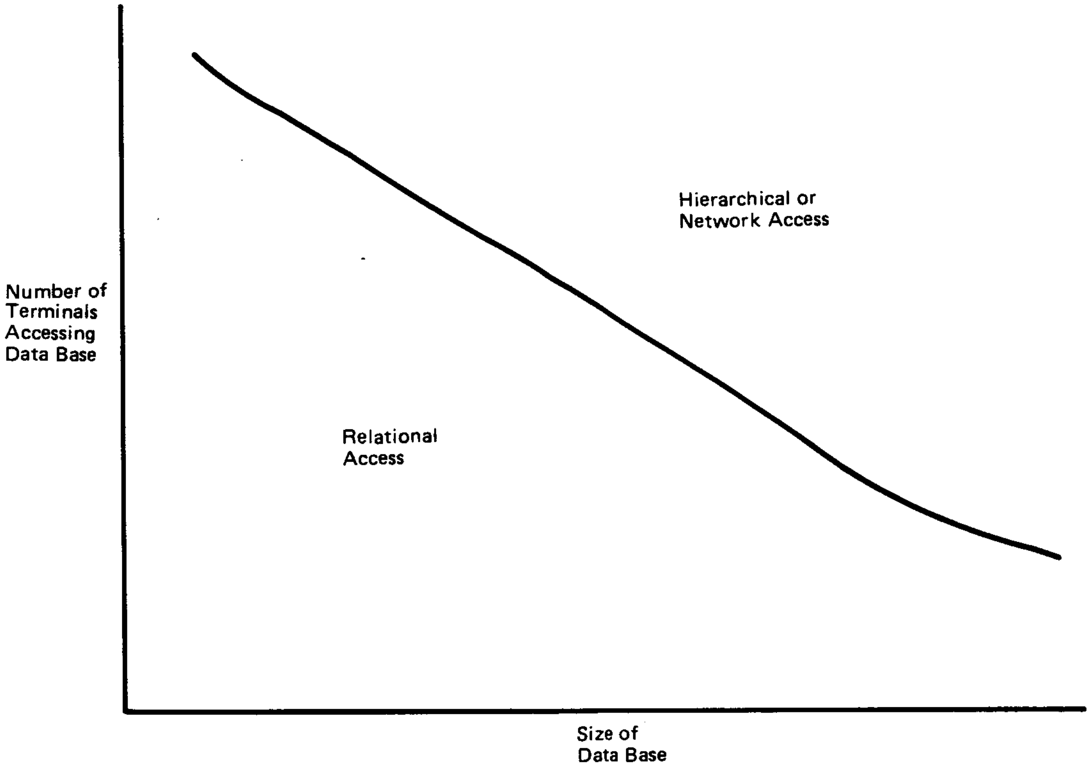

Number of
Terminals
Accessing
Data Base
Hierarchical or
Network Access
Relational
Access
Size of
Data Base

Figure 3.9.1

3.10 GROWTH ACCOMMODATION

Predicting growth in data storage is another design aspect which is critical. Large structures may become unmanageable requiring further partitioning and the data base managers must be flexible to absorb restructuring without impacting application software. Vertical growth (i.e., built-in margin) is an option and horizontal growth (i.e., expansion by adding capacity without impact to existing structures) is another option. The DB designer should factor growth into design decisions.

3.11 ARCHIVE STANDARD FORMAT (SFDU's)

The SFDU recommendations made by the CCSDS may be built into the DB archive capability. This is especially pertinent to the archiving of ancillary and scientific data. It is not clear at this time if the SFDU "labels" are related to catalogue names used by the DBMS to retrieve data blocks. Further study and decisions are required to integrate the standard formats into the DB management.

2.1.1.4. SOLUTION OPTIONS

There are three major options for data structures:

i) Relational
ii) Hierarchical
iii) Network

The benefits of relational structures are well known:

i) Easy to understand
   - tables
   - SQL language

ii) Easy to change
   - dynamic flexibility
   - continuous operation

iii) Automatic navigation
   - path selection
   - optimization

iv) Set processing

In addition, the reference configuration (see Appendix B) suggests a relational DB for the Space Station Onboard DBMS (i.e., operational data base). It seems like a good idea to use the relational model where ever possible because of the demonstrated advantages. Exceptions may include DB's only requiring file management services. The user of such services may be satisfied with and desire a hierarchical structure.

4.1 USE OF EXISTING INSTITUTIONAL RESOURCES

In any solution option the existence of planned NASA resources must be factored in. The Phase B RFP states on page C-6-5, "NASA will implement an information system for the purpose of providing the necessary collection and dissemination of data to manage the SSP". Also on page C-6-6, "Each contractor will transmit all required data to the MCDS and will receive data from the MCDS. NASA intends to utilize the MCDS throughout the SSP and to design it as the primary mechanism for routine data transfer between the contractors, NASA Level A, Level B, and Level C centers." And on page C-6-8, "NASA plans to make a data base management system and a management information system availabled on an IBM-compatible mainframe computer at JSC to support the Level B processing of management data". It is clear that the interface to and use of the MCDS(now called TMIS) must be considered in the design of all data bases defined in section 1.3.

4.2 COMMERCIAL PRODUCTS SURVEY

4.2.1 LARGE RELATIONAL PRODUCTS

Each commercial product has unique characteristics which can be used to evaluate applicability to the various data base segments. The list below contains relational products surveyed in reference 14. The complete set of characteristics is not included for all products because of the volume of material in reference 14. A table of each of the sets of characteristics which can be used in the trades are included along with an example for one system. These seem to be a comprehensive set of criteria for DBMS evaluation.

VENDOR PRODUCT CHARACTERISTICS
GENERAL INFORMATION

SYSTEM	VENDOR	CPU/OS's	COST	DEV HISTORY
SQL/DS	IBM CORP	* 370/DOS/VSE with CICS * VM/CMS		Commercial version of SYSTEM R
ORACLE	ORACLE CORP	DG/Eclipse VAX/VMS,UNIX 370-compatable /VM-CMS M68000/UNIX DEC PDP/RSTS, UNIX others		Developed as a DB manager for SEQUEL (now SQL)
INGRES	RELATIONAL TECHNOLOGY INC	VAX/VMS,UNIX M68000/UNIX others		Based on the system developed at the Univ. Calif/ Berkeley
IDM	BRITTON-LEE INC	VAX/VMS Z80/CPM Univac 1100 Datapoint 100 DG Eclipse PDP 11/UNIX		Developed as a back-end database processor using a QUEL interface
iDBP	INTEL CORP	None announced		Developed for micro and office automation appl.
RAPPORT	LOGICA LIMITED	25 mini's and mainframes		
NOMAD	D&B COMPUTER SERVICES	370-compatable/ VM/CMS MVS/TSO		Originally a reporting system for the national CSS timesharing network
ENCOMPASS	TANDEM COMPUTERS INC	TANDEM NONSTOP NONSTOP II		DBMS for a transaction processing system

SYSTEM COMPONENTS

SYSTEM	ARCHITECTURE	INTERACTIVE INTERFACE	UTILITIES	ADDITIONAL FACILITIES
SQL/DS	Two major comps RDS,DBSS	Interactive SQL(ISQL) is a CICS appl	Load/unload	Compile SQL into machine language
ORACLE	User Friendly Interface (UFI)	Interactive Application Facility(IAF)	Report writer	
NOMAD	Query,report writing etc. integrated		DBCHK to indicate DB needs to be re-organized	Bulk load

DATA STORAGE AND ACCESS TECHNIQUES

SYSTEM	DATA STORAGE	INDEX SUPPORT
SQL/DS	VSAM datasets One or more tables in VSAM dataset	B-tree indexes (no hashing) Multiple column keys Ascending or descending keys Unique keys optional Two types:clustered/non-cluster
ORACLE	Direct access files One or more tables can be stored in a file Related data from multiple tables can be stored on the same page	B-tree indexing (no hashing) Multiple column keys Ascending/descending keys Unique keys are optional
NOMAD	Data stored in direct access files	B-tree indexing (hashing) Multiple column keys Ascending/descending keys Non-unique keys optional

DATA DEFINITION

SYSTEM	TABLES	INDEXES	DATA TYPES
SQL/DS	Tables created & destroyed dynamically	Indexes created & destroyed dynamically	Character,FP,packed decimal,integer,long character Rows<4000 bytes long (excluding long character fields)
ORACLE	Tables can be created and destroyed dynamically	Indexes can be created and destroyed dynamically	Character,numeric, date,money
NOMAD	Tables can be dynamically created or destroyed	Indexes can be created and destroyed dynamically	Data types are; character,integer, FP,packed decimal, name,date

DATA MANIPULATION

SYSTEM	RELATIONAL OPERATORS	AGGREGATE FUNCTIONS	DATA INSERTION, UPDATE, DELETION
SQL/DS	Supports all (limitations on union)	Supports grouping and single level aggregation Two optimization strategies User-specified ordering	Single rows or sets
ORACLE	Uses SQL Supports all relational operators	Supports grouping and aggregation User specified ordering No optimization strategy	INSERT,UPDATE,DELETE, can be single rows or sets SQL statements for parts explosion and outer JOIN (partial)
NOMAD	Supports all relational operators	Supports grouping and aggregation User specified ordering No optimization strategy	INSERT,UPDATE,DELETE can be single rows or sets Full outer JOIN supported Statistical operators supported

DATA CONTROL

SYSTEM	AUTHORIZATION	VIEWS	SECURITY
SQL/DS	Specific authorization for-administration access, data def'n, data maniplt'n programs	Dynamic	Password optional
ORACLE	Specific authorization for: DB administration, access, data def'n, data maniplt'n programs(pre-processor chk) Decentralized authorization	Dynamic (no restrictions)	
NOMAD	Specific authorization for: access, data manipulation Centralized authorization	Dynamic view mechanism (using schemata and subschemata)	

CONCURRENCY CONTROL AND RECOVERY

SYSTEM	TRANSACTIONS	LOCKING	BACK-UP AND RECOVERY	ARCHIVAL SUPPORT
SQL/DS	User defined	Multiple level (row,page, table,DBSPACE)	Automatic for system failures	Recovery for media failures optional
ORACLE	User defined	Multiple levels of locking granularity (row,table)	Automatic	Recovery for media failures optional
NOMAD	User defined in single user mode	Single statement transactions in multiple user mode	Automatic	None

PROGRAMMING LANGUAGE INTERFACE

SYSTEM	LANGUAGES SUPPORTED	TYPE OF INTERFACE	RESTRICTIONS
SQL/DS	PL/1 COBOL FORTRAN Assembler	All SQL statements supported Multiple cursors	Limitations on host data types Program variable binding at precompilation when possible
ORACLE	COBOL FORTRAN PL/1 C	Preprocessor for COBOL Call interface for FORTRAN, PL/1 and C All SQL statements supported Allows for multiple cursors	
NOMAD	COBOL, FORTRAN, PL/1	All NOMAD commands are supported	

DOMAIN AND INTEGRITY SUPPORT

SYSTEM	DOMAINS	PRIMARY AND FOREIGN KEYS	ASSERTIONS/TRIGGERS
SQL/DS	No support	Limited support for primary keys	None
ORACLE	Limited	Limited No support of foreign keys	Assertions handled through IAF
NOMAD	Limited (primarily for report writing)	Support for primary keys No foreign keys supported	None

4.2.2 IBM PC DATA BASE PRODUCTS

The use of PC based local data base segments which are downloaded and then uploaded after updates seems to be one possible architecture for offloading the mainframe DBMS. There are over 66 database products available for the IBM PC. The PC magazine surveyed these products and reported the results in the June, July, August and September issues. The products were separated into 4 categories:

Category 1: simple file handlers operate on whole records manipulate one file at a time

Category 2: genuine data base structure (relational, network, hierarchical) manipulate two files at time query language

Category 3: capabilities of 1 and 2 procedural language

Category 4: capabilities of 3 outstanding in some way

All products within a category were given the same problem and then assessed on performance and capability. This is a fair evaluation criteria.

In category 1 the following products were recommended:

PRODUCT NAME	COMPANY	DATA MODEL	PRICE
PC-FILE III	Buttonware	Flat file	$45
Data Base Manager II	Alpha Software	File Manager	$295
ResQ	Key Software	Flat File	$295
VersaForm	Applied Software Technology	Flat File	$345
UltraFile	Continental Software	Flat File	$195
PFS:FILE	Software Publishing	Indexed File	$140
PFS:REPORT	Software Publishing	Indexed File	$125
RANK AND FILE I	RAF Software	Flat File	$160
NUTSHELL	Leading Edge Products	Flat File	$395

In category 2 the following products were recommended:

PRODUCT NAME	COMPANY	DATA MODEL	PRICE
TIM IV	Innovative Software	Flat File	$395
R:BASE SERIES 4000	Microrim	Relational	$495
PC-FILE'N REPORT	JASPIR International	Relational	$195
10 BASE	Fox Research	Relational	$495

In category 3 the following products were recommended:

PRODUCT NAME	COMPANY	DATA MODEL	PRICE
MANAGER VERSION 4	Manager Software	Relational	$195
THE SENSIBLE SOLUTION	O'Hanlon Computer Systems	Linked/indexed	$695
Condor 3	Condor Computer	Relational	$650
dBASE II VERSION2.4	Ashton-Tate	Relational	$495

In category 4 the following products were recommended:

PRODUCT NAME	COMPANY	DATA MODEL	PRICE
SAVVY PC	Excalibur Technologies	Relational	$395
Revelation	Cosmos	Relational	$950
DataFlex VERSION2	Data Access	Relational	$995
PC/FOCUS	Information Builders	Hierarchical	$1595
informix version3.11	Relational Database Systems	Relational	$795
KnowledgeMan	Micro Data Base Systems	Relational	$500
dBASE III	Ashton-Tate	Relational	$695

4.2.3 DB2

The DB2 product is the equivalent of SQL/DS but it is for the IBM MVS environment. The vendor products survey in section 4.2.1 describes SQL/DS.

4.2.4 LOCUS or DOMAIN

The LOCUS system is a UNIX based distributed system. It is described in references 3 and 4. LOCUS is a distributed operating system supporting UNIX data management facilities. Another distributed system is the Apollo Computer Inc. DOMAIN system which is also designed on an "integrated model of distributed systems". DOMAIN is described in reference 29. Both of these options are possibilities for the onboard DBMS.

4.2.5 Distributed Data Base Management Systems (DDBMS)

Distributed data base management systems are still considered to be an immature technology (see reference 19). There are many projects studying this problem. Reference 13 contains a list of projects. This appears to be a technology with potential application. This will depend on the distribution architecture of the DB segments.

REFERENCE 13: pp.62-74"Distributed Database Concepts"
Current DDBMS Projects

o Heterogeneous

COSYS	Univ. of Grenoble
CSIN	Computer Corp. of America
Sirius-Delta	INRIA- France
Sirius-POLYPHEME	Univ. of Grenoble/CII-HB
Dist. DPLS	Univ. of Tsukuba
XNDM	NBS
Multibase	Computer Corp. of America

o Homogeneous

SDD-1	Computer Corp. of America
System R*	IBM
Dist. INGRES	RTI
Honeywell-DDTS	Honeywell
XDFS	Xerox
PRIME-DTS	Prime Computer
TANDEM-ENCOMPASS	Tandem Computer

4.3 ONBOARD OPERATIONAL DATA BASE

An option that must be considered for the onboard operational DB segment is to use a commercial product. Some modification may be required if the product is not available for the processor being considered as the interface to the onboard mass storage.

Some form of a UNIX based file service may be adequate for onboard storage. The LOCUS system is a possibility (see references 2,3,4). Another possibility, which is also a distributed system, is the DOMAIN system described in reference 29. The assessment of options for the onboard DB must consider many factors:

- o communication with the ground DBMS (homogeneous/heterogeneous?)
- o performance (response,...)
- o the inherent program requirement to minimize onboard storage
- o technical issues related to autonomy and automation such as (onboard diagnostics,training,operations manuals,...)

The content of the onboard DB will be a prime driver in selecting an appropriate onboard DBMS. The TASK 1 function list suggests that the following are potential segments for residence onboard:

ONBOARD DATA BASE CONTENT

DOCUMENT MANAGEMENT
- MANUALS(PROCEDURES)
- DAILY SCHEDULES
- DIAGNOSTIC SUPPORT(SCHEMATICS)

SOFTWARE

CHECKPOINTS

SUBSYSTEM TREND DATA

REAL-TIME DATA

BUFFERED DATA(RECORDERS)

DIRECTORIES

2.1.1.5.0 OPTIONS FOR SSDS DATA BASE ARCHITECTURE

5.1 DATA BASE DISTRIBUTION AND CONNECTIVITY

The entities identified and associated connectivity are listed in the following table and shown in Figure 5.1.1.

TABLE 5.1.1 :

DATA BASE ENTITIES AND CONNECTIVITY

	HDQTRS	LVLB	LVL C				CON TRACTOR				SSE	SSCC	POCC	RDC	DDC	INTEG	TRNING	SS	COP	POP	DHC
			JSC	GSFC	MSFC	LEWI	1	2	3	4											
NASA HDQTRS		X																			
LEVEL B JSC	X		X	X	X	X															
LVL C JSC		X					X														
LVL C GSFC		X						X													
LVL C MSFC		X							X												
LVL C LEWIS		X								X											
CONTRACTOR 1			X								X										
CONTRACTOR 2				X							X										
CONTRACTOR 3					X						X										
CONTRACTOR 4						X					X										
SSE							X	X	X	X		X				X	X				
SSCC											X		X					X			X
POCC												X		X	X			X		X	X
RDC													X								X
DDC													X								X
INTEG SITE											X										X
TRAINING											X										X
SPACE STN												X	X						X		X
COP																		X			X
POP													X								X
DHC												X	X	X	X			X	X	X	

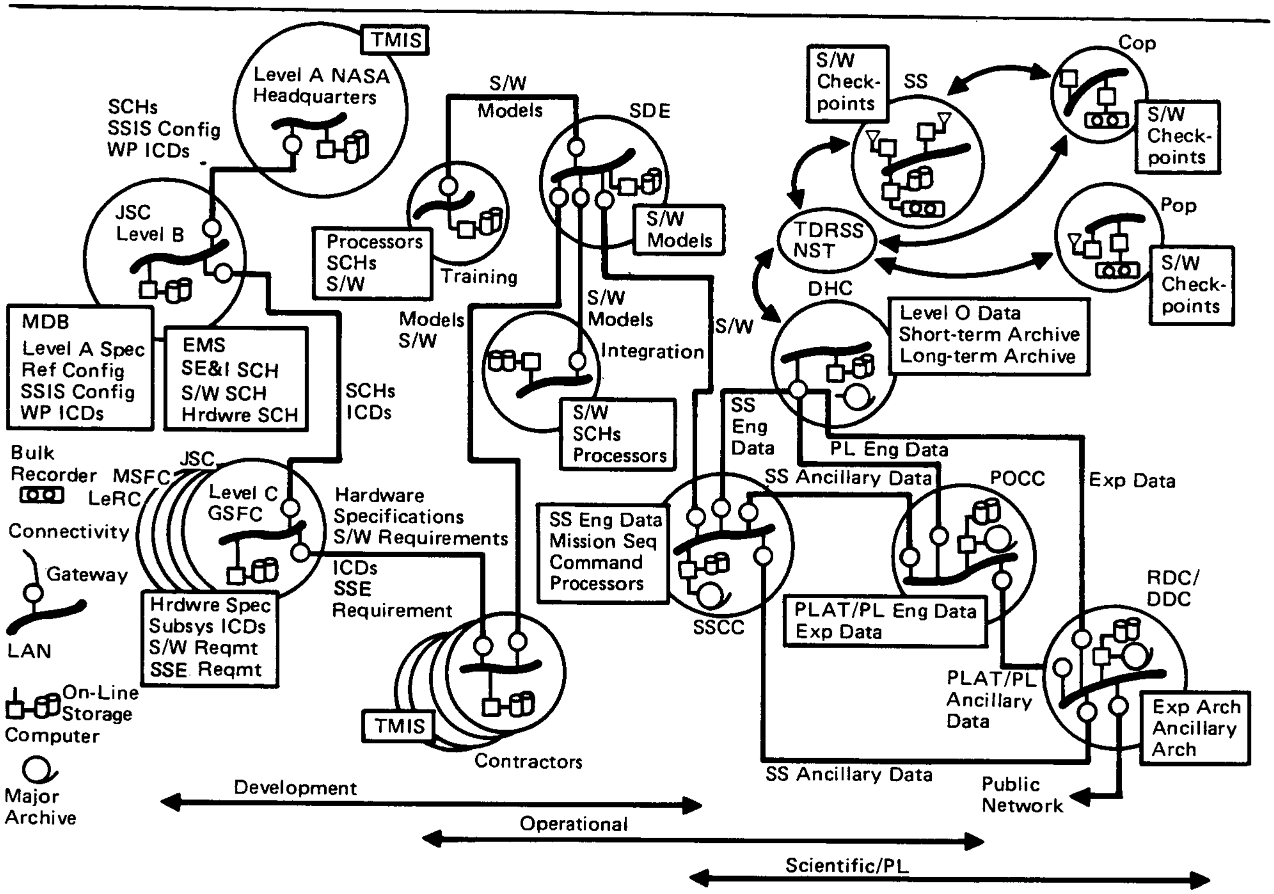


Figure 5.1.1. SSIS Data Base Distribution Option

The requirements for data base services at each entity is shown in the following table. The characterization of requirements was extracted from reference material in Appendix B and a review of features offered by various vendor products.

TABLE 5.1.2:
DATA BASE MGMT SERVICE REQUIREMENTS

	FILE SERVER								REPORTS				INTERFACE LANGUAGE						
	CREATE	STORE	RETRIEVE	COPY	MERGE	DELETE	UPDATE	TEXT PROC	CANNED	USER SPEC	PLOTS	GRAPHICS	AD HOC QUERY	INTERACTIVE	REAL TIME	PROCEDURAL	SECURITY REQD	ARCHIVE	STATISTICAL FCN
NASA HDQTRS	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
LEVEL B JSC	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
LVL C JSC	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
LVL C GSFC	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
LVL C MSFC	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
LVL C LEWIS	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
CONTRACTOR 1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
CONTRACTOR 2	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
CONTRACTOR 3	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
CONTRACTOR 4	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
SSE	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
SSCC	X	X	X										X	X	X	X		X	
POCC	X	X	X										X	X	X	X		X	X
RDC	X	X	X															X	
DDC	X	X	X															X	
INTEG SITE	X	X	X																
TRAINING	X	X	X																
SPACE STN	X	X	X										X	X	X	X			
COP	X	X	X																
POP	X	X	X																
DHC	X	X	X															X	

The data stored in various data bases within the SSP will be distributed both geographically and by interest. For example, scientific data collected from a specific experiment or PL will not be in the domain of interest of say a subsystem S/W developer who would be interested in some other part of the DB. The table below partitions the "domains of interest" into three major subdivisions. These major functional partitions and their associated "range of interest" was used in defining connectivity needed between DB's.

TABLE 5.1.3:

PARTITIONING BY MAJOR FUNCTIONAL PHASE/AREA

MAJOR PHASE/AREA	DATA APPLICATION	RANGE OF INTEREST
DEVELOPMENT	PROGRAM PLANNING	CONTRACTORS
	SOFTWARE DEV	NASA SE&I
	TEST/SIMULATION	
	TRAINING	
	SYS INTEGRATION	
OPERATIONAL	LOGISTICS PLAN	PRGM MGMNT
	MISSION PLANNING	GND CREW
	S/W RECONFIG	OB CREW
	VEHICLE MONITOR	REMOTE DISTR TO CUSTOMERS
	TRAINING	
	ANCILLARY DATA	
SCIENTIFIC/PL	RESEARCH	SCIENTISTS
	ARCHIVE	REMOTE DISTR TO CUSTOMERS

The typical data base entity will consist of the following components:

Hardware: processor(s)
storage
terminals
network interface

Software: operating system(OS)
network communication S/W
data base mgmt system(DBMS)
terminal interface S/W

A typical configuration of these components is shown in figure 5.1.2. In figure 5.1.3 and figure 5.1.4 typical configurations for commercial products are shown. These illustrate the fact that a full set of components (software and host environment) are needed to define the DBMS configuration.

5.2 APPLICABLE VENDOR PRODUCTS

All of the relational products discussed in section 4.2 are candidates for handling the large segments of the ground DB's.

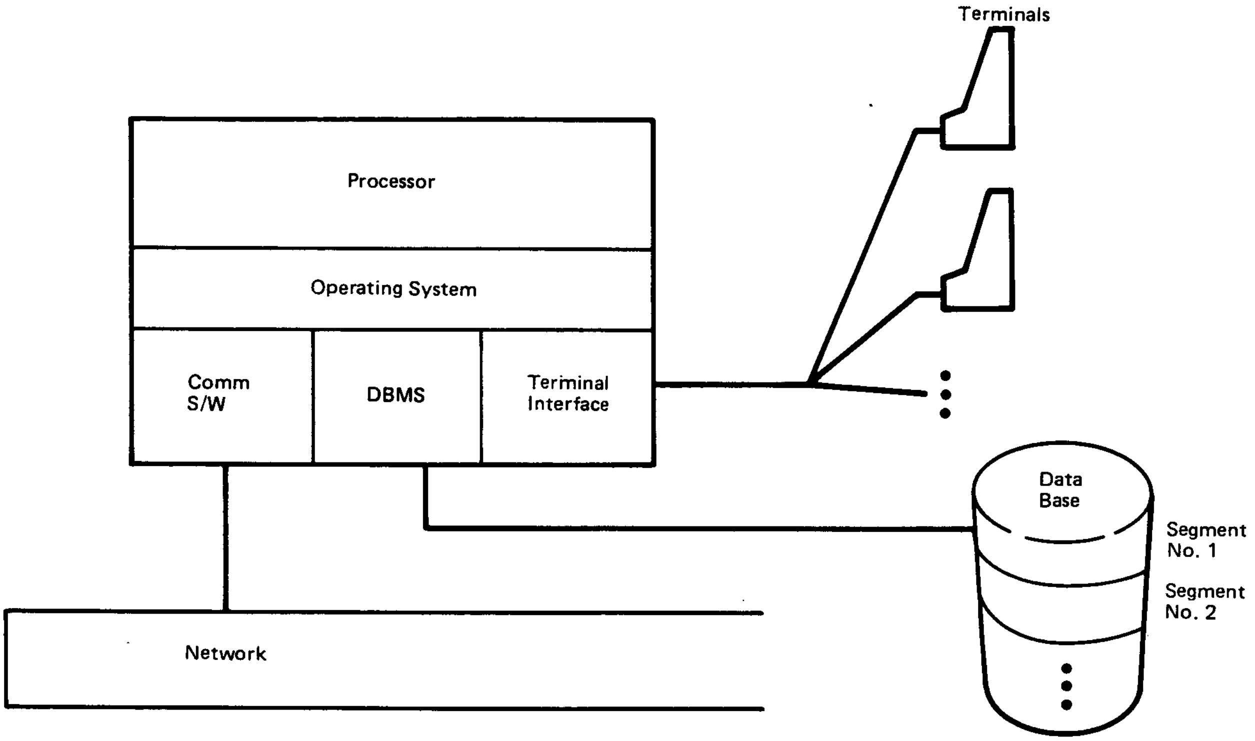


Figure 5.1.2. Typical Data Base Components

Figure 5.1.3

OPERATIONAL ENVIRONMENT

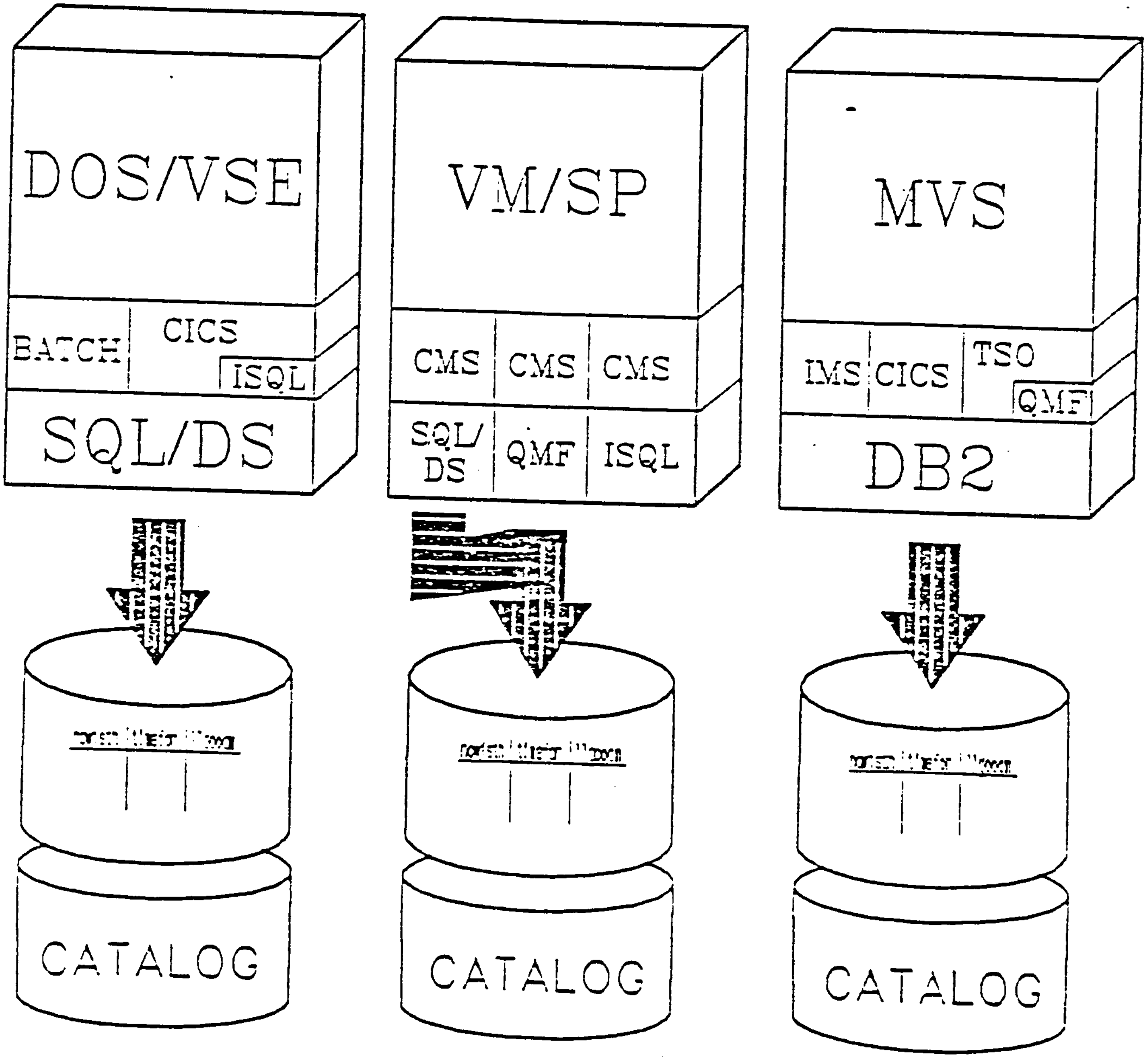
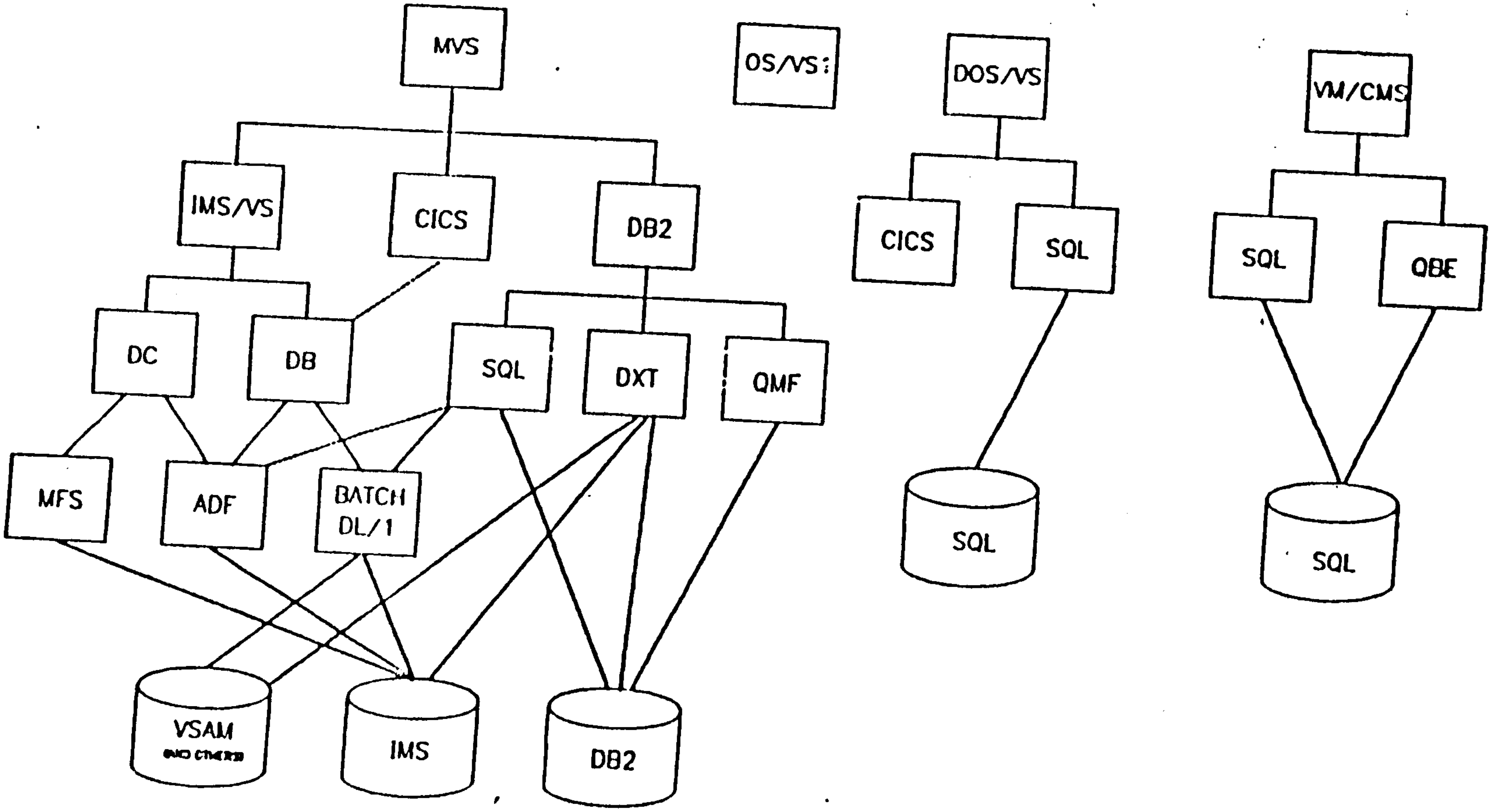


Figure 5.1.4

OPERATING SYSTEM ENVIRONMENTS



DATA BASE MANAGEMENT SYSTEMS

2.1.1.6.0 SUPPORTING MATERIAL

APPENDIX A

SSDS FUNCTIONS REQUIRING DBMS SUPPORT
(TRACEABILITY TO FUNCTION LIST)

PRECEDING PAGE BLANK NOT FILMED

SSDS FUNCTIONS REQUIRING DBMS SUPPORT

```
        1.0 Manage Customer/Operator Delivered Data
            1.1 Manage Real Time Data Return
               ______________
               |1.1.1 Capture|
               |_____________|
            1.2 Manage Delayable Data Return
               ______________
               |1.2.1 Capture|
               |_____________|
            1.3 Data Distribution
               ______________
               |1.3.2 Capture|
               |_____________|
            1.4 Manage Deliverable Customer Data
               _____________________________________
               |1.4.2 Customer Data Capture        |
               |1.4.4 Core Ancillary Data Acquistion |
               |___________________________________|
            1.5 Manage Deliverable Core Data
               __________________________
               |1.5.2 Core Data Capture |
               |________________________|

       2.0 Manage Customer/Operator Supplied Data
           ___________________________
           |2.4 Provide Ancillary Data |
           |__________________________|

       4.0 Operate Core Systems
           4.3 Support Flight Crew Activities
               4.3.5 Operations & Procedure Support
                _________________________________________
                |4.3.5.1 Maintenance & Repair Procedures |
                |4.3.5.2 Operations Procedures           |
                |4.3.5.3 Emergency Procedures            |
                |________________________________________|
           4.5 Monitor and Status System
             __________________________________________
             |4.5.4 Mass Properties Configuration Update |
             |_________________________________________|
               4.5.5 Diagnostics Support
                ________________________________
                |4.5.5.1 Schematic Presentations |
                |4.5.5.2 Expert Systems          |
                |4.5.5.3 Trend Analysis          |
                |_______________________________|
5.0 Manage Facilities and Resources
   5.1 Manage Flight System Facilities
      5.1.1 Flight Data Base Management
        ___________________________________
        |5.1.1.1 Update/Access Synch       |
        |5.1.1.2 Data File Mgmt            |
        |5.1.1.3 Mass Memory Resource Mgmt |
        |5.1.1.4 Archival Storage          |
        |__________________________________|
   5.2 Manage Space Station Control Center Facilities
      5.2.1 SSCC Data Base Management
        ___________________________________
        |5.2.1.1 Update/Access Synch       |
        |5.2.1.2 Data File Mgmt            |
        |5.2.1.3 Mass Memory Resource Mgmt |
        |5.2.1.4 Archival Storage          |
        |__________________________________|
   5.3 Manage SSDS Satellite Centers Facilities
      5.3.1 Center Data Base Management
        ___________________________________
        |5.3.1.1 Update/Access Synch       |
        |5.3.1.2 Data File Mgmt            |
        |5.3.1.3 Mass Memory Resource Mgmt |
        |5.3.1.4 Archival Storage          |
        |__________________________________|
   5.4 Manage Data Handling Center Facilities
      5.4.1 DHC Data Base Management
        ___________________________________
        |5.4.1.1 Update/Access Synch       |
        |5.4.1.2 Data File Mgmt            |
        |5.4.1.3 Mass Memory Resource Mgmt |
        |5.4.1.4 Archival Storage          |
        |__________________________________|
   5.5 Manage Development Support Facility
      5.5.1 Development Data Base Management
        ___________________________________
        |5.5.1.1 Update/Access Synch       |
        |5.5.1.2 Data File Mgmt            |
        |5.5.1.3 Mass Memory Resource Mgmt |
        |5.5.1.4 Archival Storage          |
        |__________________________________|
```

7.0 Support Space Station Program

- 7.1 Develop Logistics Plan
  - 7.1.1 Transportation Plan
  - 7.1.2 Spares and Repair Parts
  - 7.1.3 Maintenance Schedules
  - 7.1.4 Training Schedules
- 7.3 Maintain Manuals
  - 7.3.1 Maintain Operations Procedures
  - 7.3.2 Maintain Maintenance Procedures
  - 7.3.3 Maintain Satellite Services Procedures
  - 7.3.1 Maintain Emergency Procedures
- 7.4 Control Inventories
  - 7.4.1 Customer Hardware
  - 7.4.2 Station Material
  - 7.4.3 Resources
- 7.5 Configuration Management
  - 7.5.1 Interface Specifications
  - 7.5.2 System and Subsystem Specifications
  - 7.5.3 Configuration Plans
  - 7.5.4 Configuration Status

2.1.1.6.0 SUPPORTING MATERIAL

APPENDIX B

DATA BASE REQUIREMENTS DEFINITION SOURCES

This material is exerpts from the documents indicated in the headings. The paragraphs referenced are those that specify detail requirements that will drive the DBMS design.

CUSTOMER REQUIREMENTS FOR STANDARD SERVICES
FROM THE
SPACE STATION INFORMATION SYSTEM
(SSIS)

SEPTEMBER 19, 1984

pp. 2-4 para. 2.2.4

The SSIS/SCS shall implement standards at OSI layer 7 and below(where applicable) for catalogs of archived data and archived Space Station System ancillary data. As a goal, generalized standarized access protocols shall be developed for all archived data. As a minimum, a standard format shall be used to indicate the form and the format of the archived data.

pp. 3-4 para. 3.4 DATA ARCHIVE/CATALOG/RETRIEVAL

para. 3.4.1

The SSIS/SCS shall maintain data archives in two categories: customer data and engineering/ancillary data.

para. 3.4.1.1

Customer data storage shall be furthur divided into two subcategories: short term and long term. Both subcategories shall contain level0 (raw data) and level 1A data if required.

para. 3.4.1.2

Short term archives .....

para. 3.4.1.3

Long term archiving .....

para 3.4.1.4

Engineering/ancillary data (see section 4) shall be archived for a minimum of two years. .....

para. 3.4.1.5

...... The response time for average catalog query shall be comparable to similar commercial systems.

para. 3.4.1.6

.... provide the customer an inventory on what level0 and 1A data are available from the archive; .....

para. 3.4.1.7

.... maintain a log of customer data transfer requests and there disposition; ......

para. 3.4.1.8

.... to ensure the integrity of the data catalog.

para. 3.4.1.9

.... provide a system of file, catalog, index, and customer controls that will permit access to customer data. It shall include access controls to restrict access or modification to specified files, catalogs, or indexes; ......

para. 3.4.2

..... on-line data storage for customer data for 12 hours. A rapid recall off-line storage of up to 1 week ....

para. 3.4.3

.... shall provide archived data to the customer...... negotiation with customer.

para. 3.4.4

The SSIS/SCS shall make available archived data to customers in standard formats.

SPACE STATION REFERENCE CONFIGURATION DESCRIPTION

AUGUST 1984

pp. 451 para. 4.4.4 "Information and Data Management"
para. 4.4.4.1.3 "Data Protection"
para. 4.4.4.2.2.2.1 "Data Storage"
para. 4.4.4.2.2.2.4 "Data Base Management"

pp. 739 para. 5.3.6 "Information and Data Management System"
para. 5.3.6.4 "SS Reference Hardware Applicability"
para. 5.3.6.5 "Platform Reference Design"

SSIS FINAL REPORT JPL D-1737 (AUGUST 1984)

VOLUME 1 EXECUTIVE SUMMARY

pp.3-4 SERVICE REQUIREMENTS SUMMARY

Database Management Service

Provides facilities for design, generation, storage, retrieval, manipulation, and summarization of discipline oriented information files and their contents.

pp.30 INFORMATION SYSTEM SERVICES

3.8 Database Management Service

The Database Management Service provides discipline-oriented data and information bases with the following security-protected access capabilities in either interactive or batch mode:
generate and store new databases;
modify, retrieve, maintain, copy, delete, update, abstract, search, index, sort, and merge any information entities stored within the data base;
and generate reports, in canned or user-prepared formats, resulting from any of the above actions (with user interaction in interactive mode to be driven by any of three selectable options: query, menu, or command)

SPACE STATION DEFINITION AND PRELIMINARY DESIGN

REQUEST FOR PROPOSAL

(SEPTEMBER 15,1984)

pp. C-2-10 para. 3.1.3 "Command,control and Comm. Support"

"The data management services shall provide data storage, processing, presentation, and transmission services adequate to accommodate the customer requirements."

pp. C-2-18 para. 4.8.2"Data Storage"

"This parameter is the sum of the digital data storage requirement per year for each active mission." Table C-2-1 "1501.1 gigabits" (max value of data storage)

pp. C-4-31 para. 2.2.5.1b"Information security"

"The DMS shall provide data partitioning and protection to assure mission success and accommodate data privacy, commercial proprietary data, and security measures."

pp. C-4-32 para. 2.2.5.2 "Functional Requirements"

"The initial DMS shall support data base access,... on-line capabilities such as....program generation and debug, word processing, graphics, and electronic mail capabilities, health monitoring,.....display of performance and trend data, subsystem performance data, ..."

"The DMS shall support a user-friendly language for the man/machine interface. ......all phases of development and operations."

pp. C-4-33 para. 2.2.5.3 "Design and performance reqmts"

e. "Data storage and retrieval,and delivery services for for the core system data. Short term storage shall be provided onboard and selected data for transfer to archival storage on the ground. This service shall facilitate rate buffering, lifetime records. Rate buffering shall be provided for user data transmitted from the Space Station."

m. "A Data Base Management System (DBMS) shall be provided to support the convenient and effective storage, exchange, manipulation, and retrieval of data by all appropriate Space Station subsystems and users.

The DBMS software shall be capable of interfacing directly with software programs and the network operating system to ensure access to the operational data base."

n. "..... rates greater than 225 MBPS for both payloads and subsystems ..."

pp. C-6-7 Figure C-6-1 TMIS-FUNCTIONAL

```
 ________________________                      ______________________
|  LEVEL B               |                    |    NASA HDQTRS       |
|o Centralized,controlled|                    |o Level A/Level B     |
|  Data Bases            |                    |  Info Exchange       |
|o Program Mgmt Info     |                    |o Electronic Mail     |
|o SE&I Tools and Data   |<--|          |--|  |o Presentations       |
|o Centralized document  |   |          |     |o Documentation       |
|  Library               |   |          |     |                      |
|o Change Control Board  |   |          |     |                      |
|  Activities            |   |          |     |                      |
|________________________|   |          |     |______________________|
                             |          |
                             |          |
                             |          |
                             |          |
                         ____|__________|____
                        |   NETWORK          |
                        |____________________|
                             |          |
                             |          |
                             |          |
                             |          |
 ______________________      |          |      ________________________
| NASA Centers         |     |          |     | Contractor(Typical)    |
|o Level C Analysis    |     |          |     |o Analysis              |
|o Electronic Mail     |     |          |     |o Schedules             |
|o Presentations       |<----|          |--|  |o CAD/CAM/CAE           |
|o CAD/CAM/CAE         |                      |o Presentations         |
|o Documentation       |                      |o Electronic Mail       |
|o Statusing           |                      |o Documentation         |
|o Level C Data Bases  |                      |o Statusing             |
|______________________|                      |________________________|
```

6.0 SUPPORTING MATERIAL

APPENDIX C

DISTRIBUTED RELATIONAL DATA BASE
TUTORIAL
(TANENBAUM CHAPTER 10)

TANENBAUM REFERENCES:

Chu, W. W., "Optimal File Allocation in a Multiple Computer System", IEEE Transactions on Computers, Vol. C-18, pp. 885-889, Oct. 1969

Casey, R. G., "Allocation of Copies of Files in an Information Network", Proceedings SJCC, pp. 617-625, 1972

Mahmoud, S., Riordan, J. S., "Optimal Allocation of Resources in Distributed Information Networks", ACM Transactions on Database Systems, Vol. 1, pp. 66-78, March 1976

Menasce, D. A., Popek, G. J., Muntz, R. R., "A Locking Protocol for Resource Coordination in Distributed Databases", ACM Transactions on Database Systems, vol. 5, pp. 104-138, June 1980

Badal, D. Z., "On the Degree of Concurrency Provided by Concurrency Control Mechanisms for Distributed Databases", in DISTRIBUTED DATA BASES, C. Delobel and W. Litwin (eds.) Amsterdam:North Holland, pp. 35-48, 1980

Wilms, P., "Qualitative and Quantitative Comparison of Update Algorithms in Distributed Databases", in DISTRIBUTED DATA BASES, C. Delobel and W. Litwin (eds.), Amsterdam:North Holland, pp. 275-294, 1980

- o RELATIONAL DATA BASE MODEL (CODD, 1970)

  - oo INFORMATION STORED IN RECTANGULAR TABLES CALLED "RELATIONS"
    - ooo ROWS OF RELATIONS CALLED "TUPLES"
    - ooo FIELDS OF RELATIONS CALLED "ATTRIBUTES"
    - ooo COLUMNS OF RELATIONS CALLED "DOMAINS"
    - ooo DOMAIN IS SET OF VALUES FROM WHICH ATTRIBUTES ARE SELECTED

  - oo EXAMPLES OF RELATIONS

FLIGHTS*

FLIGHT#**	ORIGIN	DESTINATION	PLANE
***			
106	JFK	AMS	747
108	HPN	BOS	CNA
452	AMS	NBO	AB3
632	LHR	CDG	737
808	SFO	JFK	D10
.	.	.	.

RESERVATIONS

DATE	FLIGHT #	BOOKED	NAME
29 MAR	106	210	BARBARA BONGO
21 FEB	108	005	CAROL CURLEW
16 MAR	632	105	MARIA MARMOT
01 SEP	108	000	BARBARA BONGO
.	.	.	.
.	.	.	.

*RELATION **DOMAINS (OR ATTRIBUTES) ***TUPLES

PERSONNEL

NAME	TITLE	SEX	MARRIED	SENIORITY
MARILYN MANATEE	MANAGER	F	N	4
SUZANNA SPRINGBOK	PUBLICIST	F	Y	3
BARBARA BONGO	PILOT	F	N	3
MARVIN MOONFISH	ENGINEER	M	N	6
ANDREW AARDVARK	PROGRAMMER	M	Y	10
.	.	.	.	.

AIRCRAFT

PLANE	SEATS	ENGINES
747	450	4
737	100	2
D10	270	3
AB3	260	3
CNA	4	1
.	.	.

DICTIONARY

RELATION	LOCATION	# TUPLES	TUPLE SIZE	# DOMAINS
FLIGHTS	AMSTERDAM	1,000	12	4
RESERVATIONS	NEW YORK	365,000	30	4
PERSONNEL	NAIROBI	10,000	35	5
AIRCRAFT	MELBOURNE	10	8	3
.	.	.	.	.

oo EACH RELATION HAS A "KEY"

ooo KEY IS ONE OR MORE DOMAINS THAT UNIQUELY DETERMINE VALUE OF THE DOMAIN

ooo FOR FLIGHTS THE FLIGHT # IS THE KEY DOMAIN

ooo FOR RESERVATION THE KEY IS THE DATE AND FLIGHT # DOMAINS

oo RELATIONS CAN ALSO BE STORED AS A SPECIAL RELATION CALLED THE "DICTIONARY"

ooo USED TO FIND WHERE RELATION IS STORED

oo DATA BASE MAY BE "FULLY PARTITIONED" OR "FULLY REPLICATED" OR SOMETHING BETWEEN

ooo "FULLY PARTITIONED" - EACH RELATION STORED IN EXACTLY ONE LOCATION

ooo "FULLY REPLICATED" - EACH RELATION STORED IN ALL LOCATIONS

oo REPLICATION IS NOT DESIREABLE IF UPDATES ARE COMMON

ooo DIFFICULTY IN KEEPING COPIES IDENTICAL

ooo NETWORK DELAYS

o QUERY LANGUAGES

oo ALL DATA BASES ALLOW "QUERIES" AND "UPDATES"

oo FOR RELATIONAL DATA BASES THERE ARE TWO CATEGORIES OF QUERY LANGUAGE

ooo PROCEDURAL OR "RELATIONAL ALGEBRA"
ooo NON-PROCEDURAL OR "RELATIONAL CALCULUS"

oo THREE MAJOR OPERATIONS: RESTRICTION, PROJECTION, JOIN

ooo "RESTRICTION" TAKES RELATION AND "PREDICATE" (ARBITRARY BOOLEAN EXPRESSIONS ON VALUES OF DOMAIN) AND PRODUCES SUBSET OF TUPLES MEETING PREDICATE
E.G., RELATION PERSONNEL, PREDICATE SEX = 'M'

ooo "PROJECTION" TAKES TWO RELATIONS AND A COMMON DOMAIN AND PRODUCES NEW RELATION WITH SPECIFIED DOMAINS

ooo "JOIN" TAKES TWO RELATIONS AND A COMMON DOMAIN AND PRODUCES A NEW RELATION MERGED ON COMMON DOMAIN
E.G., FLIGHTS AND AIRCRAFT COULD HAVE "PLANE" AS JOINING DOMAIN

o USE OF QUERY LANGUAGES

oo RELATIONAL ALGEBRA QUERY EXAMPLE - LIST ALL DATES WHERE THERE ARE OVERBOOKED FLIGHTS FROM BOS TO SFO

ooo ALGEBRA QUERY GIVES INSTRUCTIONS ON HOW TO GET ANSWERS

```
T1 = RESTRICTION OF FLIGHTS BY ORIGIN = BOS AND DESTINATION = SFO
T2 = JOIN OF T1 AND AIRCRAFT ON PLANE
T3 = PROJECTION OF T2 ONTO FLIGHT # AND SEATS
T4 = JOIN OF T3 AND RESERVATIONS ON FLIGHT #
T5 = RESTRICTION OF T4 BY BOOKED > SEATS
```

oo RELATIONAL CALCULUS QUERY USES "<u>TUPLE VARIABLES</u>" TO SPECIFY CONDITION RESULT MUST SATISFY

ooo TUPLE VARIABLE RANGES OVER ONE RELATION - E.G., F IS TUPLE VARIABLE OVER FLIGHTS FROM F.ORIGIN IS DOMAIN OF FLIGHTS

ooo SAME PROBLEM AS ABOVE IN RELATION CALCULUS:

```
RETRIEVE R.DATA

   WHERE
        R.FLIGHT # = F.FLIGHT # AND F.PLANE = A.PLANE
   AND
        F.ORIGIN = BOS AND F.DESTINATION = SFO
   AND
        R.BOOKED > A.SEATS
```

ooo CONCEPTUALLY, QUERY CAN BE ANSWERED BY THREE NESTED LOOPS, ONE ON EACH TUPLE VARIABLE - INSPECT EACH COMBINATION OF THREE TUPLES (ONE FROM R, F, A RELATIONS) TO SEE IF IT MEETS PREDICATE.

ooo TEST WOULD BE MADE 1,000 x 365,000 x 10 TIMES

oo DATA BASE SYSTEM MUST AVOID IMPOSSIBLY LONG BRUTE FORCE SEARCHES

o PROBLEMS INTRODUCED BY DISTRIBUTING DATA BASE
  (1) WHERE TO PUT RELATIONS.
  (2) HOW TO PROCESS QUERIES?
  (3) HOW TO KEEP MULTIPLE TRANSACTIONS FROM INTERFERRING?
  (4) HOW TO MAINTAIN INTEGRITY WHEN SYSTEM CRASHES OCCUR?

o DISTRIBUTION PROBLEM
  oo DATA BREAK DOWN TO HOSTS (SUBDIVIDE RELATIONS?, SMALLER FRAGMENTS?)
  oo RELATION DISTRIBUTION TO HOSTS? (TRADES)
  oo ASSUME FULLY PARTITIONED DATA BASE AND EACH RELATION COMPLETELY STORED AT A SINGLE HOST
  oo DICTIONARY RELATION LIKELY WILL BECOME BOTTLENECK SO REPLICATE AT EACH HOST (SELDOM CHANGES)
  oo PROBLEM HAS BEEN SOLVED FOR FULLY CONNECTED FULL-DUPLEX NETWORK WHERE PRINCIPAL TRAFFIC IS DUE TO QUERY RESPONSE AND DELAY DUE TO QUEUEING (CHU 1969)
  oo SOLUTION IS AN ASSIGNMENT OF ONES AND ZEROS TO THE MATRIX X
  oo CHU FINDS ANALYTIC EXPRESSION FOR qij USING QUEUEING THEORY BUT SOLUTION USES ZERO-ONE LINEAR PROGRAMMING (CASEY 72', MAHMOND, RIORDAN 76')

GIVEN:

n = NUMBER OF HOSTS

m = NUMBER OF RELATIONS

b_i = STORAGE CAPACITY (IN BYTES) AT HOST i

c_i = COST OF STORAGE (PER BYTE PER SECOND) AT HOST i

L_j = SIZE OF RELATION j

l_j = TUPLE SIZE (IN BYTES) FOR RELATION j

r_{ij} = TUPLES/SEC FROM HOST i FOR RELATION j

T_{ij} = MAXIMUM ACCEPTABLE DELAY FOR HOST i TO A RELATION j TUPLE

C_{ik} = COMMUNICATION COST (PER BYTE) FOR TUPLES FROM i TO k

q_{ij} = DELAY FOR QUERY FROM i TO j

VARIABLES:

X_{ij} = 1 IF RELATION j IS AT HOST i, 0 OTHERWISE

CONSTRAINTS:

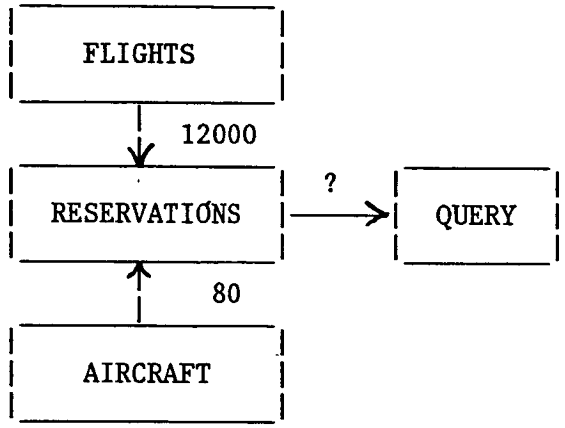
$$\sum_{i=1}^{n} X_{ij} = 1 \qquad (1 \leqq j \leqq m)$$

$$\sum_{j=1}^{m} X_{ij} L_j \leqq b_i \qquad (1 \leqq i \leqq n)$$

$$q_{ij} \leqq T_{ij} \qquad (1 \leqq i \leqq n,\ 1 \leqq j \leqq m)$$

GOAL: MINIMIZE TOTAL COST $= \underbrace{\sum_{i=1}^{n} \sum_{j=1}^{m} X_{ij} c_i L_j}_{\text{STORAGE COST PER SECOND}} + \underbrace{\sum_{i=1}^{n} \sum_{k=1}^{n} \sum_{j=1}^{m} X_{kj} C_{ki} r_{ij} l_j}_{\text{COMMUNICATION COST PER SECOND}}$

- QUERY PROCESSING PROBLEM
  - TWO POSSIBLE METRICS TO MEASURE GOODNESS OF A QUERY DISTRIBUTION STRATEGY
    (1) QUERY RESPONSE TIME (INTERACTIVE APPLICATIONS)
    (2) TOTAL BANDWIDTH CONSUMED (BATCH TRANSACTION PROCESSING)
  - SIMPLIFYING ASSUMPTION - REASONABLE TO ASSUME COST AND TIME TO MOVE DATA IS SAME FOR ALL PAIRS OF HOSTS
  - STRAIGHT FORWARD APPROACH
    - QUERIED HOST COMMANDS OTHER HOST TO SEND REQUIRED RELATIONS
    - QUERY HOST DOES PROCESSING
    - ALWAYS WORKS BUT THERE ARE MORE EFFICIENT SOLUTIONS
  - AVOID MOVING LARGE RELATIONS
    - IN EXAMPLE CONSIDERED ABOVE QUERY HOST COMMANDS FLIGHT HOST TO SEND RELATION TO RESERVATIONS HOST AND ALSO AIRCRAFT HOST TO SEND RELATION TO RESERVATION HOST



  - QUERY HOST DOES NOT HAVE ENOUGH INFORMATION TO MAKE OPTIMAL DECISION
  - CAN STILL DESIGN ALGORITHMS THAT GIVE OPTIMAL PERFORMANCE UNDER CERTAIN STATISTICAL CONDITIONS (CHU, HURLEY 1979)
  - DO PROCESSING BEFORE RELATIONS ARE MOVED
    - QUERY HOST BROADCASTS QUERY TO ALL RELEVANT HOSTS
    - LOCAL PROCESSING DONE ON RELATIONS (RESTRICT TO DOMAIN VALUES IN QUERY)
    - PROJECTIONS DONE LOCALLY (EXAMPLE CONSIDERED DOESN'T NEED A.ENGINES, F.PLANES SO DON'T SEND)

o CONCURRENCY CONTROL PROBLEM

oo CONCURRENCY CONTROL ALGORITHMS MUST MAXIMIZE PARALLEL ACTIVITY WHILE MAINTAINING DATA BASE CORRECTNESS

oo DEFINE "TRANSACTION" AS SET OF READS
$R = R(x_1, x_2, \dots x_n)$ AND SET OF WRITES

$$w = w(y_1, y_2, \dots y_n)$$

oo DEFINE A "LOG" AS TIME ORDERED SEQUENCE OF READS AND WRITES

oo EXAMPLES FOR TWO TRANSACTIONS

$$LOG_1 = R_1W_1R_2W_2$$

$$LOG_2 = R_2W_2R_1W_1$$

$$LOG_3 = R_1R_2W_1W_2$$

oo LOG IS "SERIAL" IF EACH R_i IS IMMEDIATELY FOLLOWED BY W_i
OTHERWISE IT IS "INTERLEAVED"

oo INTERLEAVED LOGS DO NOT NECESSARILY PRODUCE CORRECT RESULTS

oo IF INTERLEAVED LOG CAN BE SHOWN TO BE EQUIVALENT TO SOME SERIAL LOG ("SERIALIZATION") THEN IT PRODUCES CORRECT RESULTS (E.G., $R_iW_j = W_jR_i$ IF OPERATE ON DISJOINT SETS OF DATA; READS CAN BE REORDERED OR PERFORMED IN PARALLEL)

oo MOST ALGORITHMS ACHIEVE SERIALIZABLE LOGS BY ALLOWING TRANSACTIONS TO "LOCK" PART OF DATA BASE BEFORE STARTING

oo SIMPLIEST LOCKING SCHEME - SINGLE LOCK OF ENTIRE DATA BASE (ONLY ONE TRANSACTION AT A TIME CAN RUN)

oo COULD ALSO LOCK RELATIONS

oo OTHER POSSIBILITIES
(1) LOCK INDIVIDUAL TUPLES
(2) LOCK INDIVIDUAL DOMAINS
(3) LOCK INDIVIDUAL VALUES
(4) LOCK PHYSICAL DISK SECTORS

oo ASSUME RELATIONS ARE UNIT OF LOCKING

oo LOCKS MUST BE ACQUIRED SEQUENTIALLY

oo "DEADLOCK" SITUATION CAN RESULT (E.G., HOST A LOCKS R_1,
HOST B LOCKS R2, EACH HOST NOW ATTEMPTS TO LOCK OTHER RELATION)

oo "TWO PHASE LOCKING" PROTOCOL SOLVES PROBLEM

oo HOSTS GO THROUGH PHASE OF ACQUIRING LOCKS AND THEN A PHASE OF RELEASING LOCKS; ONCE SOME LOCK HAS BEEN RELEASED NO OTHER LOCKS CAN BE ACQUIRED o DISTRIBUTED DATA BASE CONCURRENCY CONTROL

oo ILLUSTRATED BY ONE POSSIBLE ALGORITHM (SIMPLE EXAMPLE BUT DOES NOT ALLOW MAXIMAL CONCURRENCY)

oo ALGORITHM BASED ON ASSIGNING UNIQUE NUMBER TO TRANSACTIONS

TIME	HOST

UNIQUE TRANSACTION NUMBER

oo CLOCKS OF HOSTS NEED NOT BE SYNCHRONIZED

oo EACH HOST MAINTAINS FOUR VARIABLES:
t = NUMBER OF CURRENT TRANSACTION
n = NUMBER HOSTS IN TRANSACTION
k = LOCK COUNTER
s = CURRENT STATE (IDLE, LOCKING, PROCESSING, UNLOCKING)

oo QUERY ARRIVING AT HOST IN IDLE PROCEEDS AS FOLLOWS:
(1) SWITCH TO LOCKING STATE
(2) SET $k = o$
(3) BROADCAST PLEASE_LOCK MESSAGE TO $n-1$ HOSTS (MESSAGE CONTAINS TRANSACTION NUMBER, ETC.)
(4) WAIT FOR $n-1$ PLEASE_LOCK MESSAGE TO COME BACK

oo QUERY ARRIVING AT HOST NOT IN IDLE IS QUEUED

oo PLEASE_LOCK MESSAGE ARRIVAL AT HOST IN IDLE PROCEEDS AS FOLLOWS:
(1) SWITCH TO LOCKING STATE
(2) SET $k = 1$
(3) SET $t = 1$ MESSAGE TRANSACTION NUMBER
(4) SET n = NUMBER OF HOSTS IN TRANSACTION (IN MESSAGE)
(5) REBROADCAST MESSAGE TO $n-1$ HOSTS

oo PLEASE_LOCK MESSAGE ARRIVAL AT HOST IN LOCKING STATE PROCEEDS AS FOLLOWS:
(1) CHECK TRANSACTION NUMBER
(2) IF TRANSACTION NUMBER LOWER THAN t THEN QUEUE PREVIOUS TRANSACTION AND START OVER WITH NEW TRANSACTION AS IF PLEASE-LOCK ARRIVED WITH HOST IN IDLE STATE
(3) IF TRANSACTION NUMBER = t THEN INCREMENT k
(4) IF TRANSACTION NUMBER > t THEN IGNORE

oo WHEN $k = n-1$ (ALL HOSTS HAVE SENT PLEASE_LOCK MESSAGE) THEN HOST CAN BEGIN PROCESSING AS FOLLOWS:
(1) QUERY PROCESSING ON RELATIONS
(2) WHEN PROCESSING COMPLETE THEN SEND $n-1$ HOSTS PLEASE_UNLOCK MESSAGE
(3) WHEN $n-1$ RETURN MESSAGES RECEIVED THEN HOST SWITCHES TO IDLE

oo BETTER ALGORITHMS PROPOSED
(1) MENASCE ET AL 1980
(2) BADAL 1980
(3) WILMS 1980

2.1.1.7.0 GLOSSARY OF TERMS

ad-hoc query: a request for information made to a database system that was not specifically planned for and programmed into the system

Atomic: non-decomposable so far as the system is concerned

Audit trail: a report or file generated by a DBMS that records transactions that add to,change,or delete from the DB

B-tree index: short for balanced tree, a way of organizing the pointers to information in DB's that allows quick retrieval of any single specified record; The combination of the "index set" and the "sequence set" defined in VSAM is sometimes called a B-tree

CODASYL: an acronym for Conference On Data Systems and Languages a federally sponsored industry committee that developed standards which led to the COBOL language and many of the more complex types of data bases

Hashing: replacement of a sequential search by an address look-up; the address look-up is through an address function called the hashing function

Key: each relation has a key consisting of one or more domains that uniquely determine the values of the other domains

Navigation: the process of traveling through the DB, following explicit paths from one record to the next in search for some required piece of data

Normalization: the process that results in an end file design where only data about the prime key is in each relation

Stored record address(SRA): a unique value created by the access method to distinquish the stored record occurrence from all others in the DB (e.g. the physical address of the occurrence in the storage volume)

View: a way of presenting the contents of a DB to the user, not necessarily the same as the way the fields and records are stored in the DB

VSAM: Virtual System Access Method consisting of two index sets; The "sequence set" is a single-level dense index to the actual data; provides fast sequential access The "index set" is a tree structure to the "sequence set"; provides fast direct access to the "sequence set";

2.1.1.8.0 REFERENCES

1. Andrew S. Tanenbaum, "Computer Networks", Prentice Hall, 1981 pp.440 section 10.1 Distributed Data Base Systems

2. Brian W. Kernighan, John R. Mashey, "The UNIX Programming Environment", Software-Practice and Experience, Vol.9, pp. 1-15,1979

3. G. Popek, B. Walker, J. Chow, D. Edwards, C. Kline, G. Rudsin, G. Thiel, "LOCUS: A Network Transparent, High Reliability Distributed System", Operating Systems Review 15(5), pp. 169-177, 1981 Proceedings ACM 8th Conf. Operating Systems Principles, Asilomar, Calif.

4. Walker B., Popek G., English R., Kline C.,Thiel G., "The LOCUS Distributed Operating System", Operating Systems Review, pp. 49-69, 1983 Proceedings ACM 9th Conf. Operating Systems Principles

5. Consultative Committee For Space Data Systems (CCSDS), Panel 2: Standard Data Interchange Structures, "CCSDS Reference Document On Space Data Systems Operations With Standard Data Units: System and Implementation Aspects", Issue 1, June 1984

6. CCSDS Panel 2: Standard Data Interchange Structures, "Recommendation For Space Data System Standards: Standard Format Data Units-- Concept and Primary Label", Draft 3, July 1984

7. Shrivastava, Santosh Kumar, "Structuring Distributed Systems for Recoverability and Crash Resistance", IEEE Trans. on Software Engineering, Vol. SE-7, No.4, July 1981

8. Jensen E. Douglas, "The Archons Project: An Overview", Carnegie-Mellon University, Revised March 22 1983

9. Space Station Information System (SSIS) Final Study Report Volume 1, Executive Summary JPL D-1737, August 1984

10. Brownbridge D.R., Marshall L.F., Randell B., "The Newcastle Connection or UNIXes of the World Unite!", Software-Practice and Experience, Vol. 12, pp. 1147-1162 1982

11. Erdogan S.S., "Archive Management in Distributed Systems", Proceedings IEEE INFOCOM 84, pp. 398-411, April 1984

12. Date C.J., "An Introduction to Database Systems", Addison Wesley, 3rd Edition, 1981

13. Schmidt, Noel, "Local Computer Networks: Software Applications", Tutorial 4, IEEE INFOCOM 84

14. David Upham, Technology Transfer Institute, "Relational Data Base 3-Day Seminar"

15. Smith T.J., "Space Station Flight Data System: Data Acquisition, Telecommands, and Packet Telemetry", May 21, 1984, IBM FSD report

16. Berstein, P.A., Goodman, N., "Concurrency Control in Distributed Database Systems", ACM Computing Surveys, Vol.13, No.2, June 1981

17. Codd, E.F., "A Relational Model of Data for Large Shared Data Banks", Communications ACM, Vol.13, pp. 377-387, 1970

18. Proceedings: The 1984 National Database and Fourth Generation Language Symposium, Digital Consulting Associates, Inc., Fall 1984 Edition

19. Information System News, November 12, 1984, "Distributed DBMS Facing Several Technological Hurdles", pp. 24-30

20. Larry R. Harris, "Experience with INTELLECT Artifical Intellegence Technology Transfer", The AI Magazine, pp. 43-50, Summer 1984

21. GSFC Space Station Office,"Customer Requirements For Standard Services from the Space Station Information System (SSIS)", September 19, 1984, Table 4.1 pp. 4-2

22. Badal, D.Z., McElyea, W., "A Robust Adaptive Concurrency Control For Distributed Databases", Proceedings IEEE INFOCOM 84, pp. 382-391, April 1984

23. Gopal, I.S., Segall, A., "Dynamic Address Assignment Protocols", Proceedings IEEE INFOCOM 84, pp. 120-128, April 1984

24. Sproull, F.S., Cohen,D., "High Level Protocols", Proceedings IEEE, vol. 66, No. 11, pp. 1371-1386 November 1978

25. Day, D.D., "Resource Sharing protocols", IEEE Computer, pp. 47-56, September 1979

26. Day, D.D., "Terminal Protocols", IEEE Transactions on Communications, vol. COM-28, No. 4, pp. 585-593, April 1980

27. Chou, C.P., Lui, M.T, "A Concurrency Control Mechanism and Crash Recovery for a Distributed Database System", in DISTRIBUTED DATA BASES, C. Delobel and W. Litwin (eds.) Amsterdam:North Holland pp. 201-214, 1980

28. Lampson, B.W., "Atomic Transactions", in Distributed Systems An Advanced Course. Berlin:Springer-Verlag, 1980

29. Leach, Paul J., Leven, Paul H., Douros, Bryan P., Hamilton, James A., Nelson, David L., Stumpf, Bernard L., "The Architecture of an Integrated Local Network", IEEE Journal on Selected Areas in Communications, Vol. SAC-1, No. 5, November 1983

SSDS COMMAND MANAGEMENT AND RESOURCE MANAGEMENT

2.1.2.1.0 INTRODUCTION

1.1 Purpose

The primary purpose of this activity is to identify options for the implementation of functional requirements, as shown in [1], for two (2) major functions: Function 2.0 Manage Customer/Operator Supplied Data, and function 3.0 Schedule and Execute Operations. In the areas of command management and resource management, some key options will be identified. Major issues associated with each option will be briefly described and key options characterization will be discussed.

A command is defined as a set of instructions which cause a payload or core system to execute a specific operation or set of operations. Commands are separate from (and expected to be a small part of) the general data uplink.

Resources are defined as those common commodities and services available for use and shared among payloads and SSDS systems. Resources of special inportance onboard the spacecraft include electrical energy, thermal heat rejection, local area network and communications bandwidth, and crew operator time. Ground resources of special importance include mass storage for data capture and archive, level 0 processors, and communications bandwidth, as well as facilities supporting development integration and checkout of hardware and software.

While the primary thrust of the paper is to the Space Station, it is anticipated that the options described will also support platform command and resource management. The specific modifications necessary include:

- o No onboard customer or operators.
- o Higher degree of onboard automation essential.
- o Greater emphasis on integrated, planned operation, with probable use of expert systems for scheduling.

1.2 Approach

Options for command and resource management will be developed and characterized through the following five steps:

- Establish a context for the options. A majority of the context for command and resource management derives directly from requirements specified in the GSFC "Customer Requirements for Standard Services" (CRSS) document of Reference 2. The remainder of the context derives from Space Station requirements [3], mission requirements [4], MDAC analyses, and other NASA and contractor analyses. The context serves to limit the scope of options which may be considered, and to identify the events which must take place to satisfactorily perform these functions. However, MDAC understands that attractive options should not be automatically excluded because they fail to meet all requirements and criteria. NASA should have the opportunity to modify the requirements to admit effective solutions.

- Identify the full range of attractive system concept options. This step will be accomplished first at the system level to ensure that there is no inadvertent failure to identify promising overall approaches.

- Define a reference system. The detailed steps in command and resource management must be identified in order to show the complete functioning of the system. However, options of major importance will only exist for some of these steps. The implementation of the remainder will be left for Task 4. The reference system also serves a second, very important role. When properly described, the system will be sufficiently flexible to intrinsically contain the full range of system level options. The options development may then be combined to synthesize a broad range of potential methods for command and resource management.

- Identify key option areas. Those steps which are keys in defining the command and resource methodology will be identified in Sections 2.1 and 3.1. These options will be selected based on their role in establishing the nature of the command and resource management functions. Steps not selected for options development will also be shown for comparison.

- Develop and characterize options. Options will be defined to cover the attractive approaches to accomplishing each selected step. These options will reflect current and recent NASA programs and planning, as well as the MDAC team analyses and concepts. Characterization will include meeting requirements, resolving key issues, and applicable aspects of cost, performance, risk, and convenience.

1.3 Context, Concepts and Reference Systems

These first three steps in the methodology are still introductory in nature, as they establish the scope of the investigation.

1.3.1 Context for Command and Resource Management

The command management function must include the steps shown in Table 1.3-1. Variations in the sequence of the steps and the routing of commands through the steps are permissable.

The table shows the steps and one or more requirements that the step will completely or partially fulfill. Note that these requirements do not necessarily demand each of these steps. However, the MDAC system functional analysis has confirmed the importance of each step.

Requirements which provide direction on the nature of the command management function are shown in Table 1.3-2. This listing combines essential features of one or more representative requirements into a single summary statement. The design of the function should consider the complete text of all applicable requirements. A listing of these requirements is provided in Reference 1.

Table 1.3-1

Essential Steps for Command Management

STEP	SOURCE
o Authenticate Command Sender and Address	CRSS[2] 1.1.4
o Determine Command Classification (Restricted, Constrained, or Non-restricted)	CRSS 6.1.6
o Pass non-restricted commands and data to their destination with no further checking.	CRSS 6.1.6.3
o Determine whether restricted and constrained commands are executable.	CRSS 6.1.6.2
o Pass executable, restricted/constrained commands to their destination at appropriate times.	CRSS 6.1.3.1 CRSS 7.3.4.1
o Attempt to resolve problems with not-executable commands.	
o Return not-executable commands to sender.	CRSS 7.3.4.1
o Report all command disposition and status to sender.	CRSS 7.3.4.1
o Provide mechanism for customer to cancel commands.	

2. Customer Requirements for Standard Services, Reference 2

Table 1.3–2

Requirements for Command Management

STEP	SOURCE
o Provide for Command Data Privacy.	RFP[3] C–4–2.2.6.2t CRSS[2] 5.4.4 RFP C–3–3.1.d RFP C–3–3.1.A CRSS 2.2.2
o Process all commands in a manner consistent with Customer Real Time, Interactive Operation.	CRSS 1.1.9 CRSS 6.2.1 CRSS 6.2.1.2 CRSS 6.2.1.3 CRSS 6.1.5
o Support generation and real time change of stored command sequence.	CRSS 6.2.1.4 CRSS 6.3.1 CRSS 6.1.2 CRSS 6.3.2
o Support Customer Payload Commanding	CRSS 6.1.1
o Make Command Entry and Resolution User Friendly.	CRSS 1.1.4 CRSS 1.1.8
o Enable Customer Payload Control to be essentially the same as if the payload were in his own laboratory.	CRSS 6.1.3
o Accept commands originated by the payload and onboard core systems.	CRSS 6.3.1 CRSS 6.3.2

[2] Customer Requirements for Standard Services, Reference 2
[3] Space Station RFP Section C, Reference 3

Essential steps for resource management are shown in Table 1.3-3. Note that one step is not supported by a specific requirement. MDAC has assumed that Space Station program major events such as NSTS launches, payload additions and removals, maintenance actions on free fliers such as the Hubble Space Telescope, OTV launches, etc. will be planned and scheduled by the Space Station program and provided to the SSDS as a skeleton around which to build the operating schedules.

Resource management requirements summary statements are shown in Table 1.3-4. As with command management requirements, the complete text of all applicable requirements should be used to design the function.

1.3.2 System Options and Reference System for Command Management

The two extremes of command management are illustrated in Figures 1.3-1 and 1.3-2. The system in Figure 1.3-1 places responsibility for classifying commands as to restricted, constrained, and non-restricted on the customer or payload. Conditions for executability of constrained commands are determined by the customer, and non-restricted and executable, constrained commands are sent to the payload. The payload may execute self generated, non-restricted commands without any checking. The SSDS has no role in processing these customer and payload initiated commands. SSDS responsibility is limited to determining the executability of customer initiated restricted commands and payload initiated restricted and constrained commands. Those executable are sent to the payload. Those not executable are rejected. The determination of executability depends only on impacts on the Space Station safety. Since there is no command dictionary or equivalent method of determining executability of a top level in this system the SSDS must read and understand the command in order to determine executability.

The system illustrated in Figure 1.3-2 places minimal responsibility on the customer. The SSDS receives and processes all commands from the customer and restricted and constrained commands from the payload. Those commands unknown to the SSDS are rejected. Those found executable, including non-restricted commands, are delivered to the payload. Those not executable are returned to

Table 1.3-3

Essential Steps for Resource Management

STEP	SOURCE
o Accept and verify operations requests from customers and station operators.	CRSS[2] 8.4.2 CRSS 7.2.2
o Receive and confirm Major Event SSP from SSP.	MDAC Analysis
o Negotiate Communications Requirements with NCC.	CRSS 5.1.4 CRSS 5.2.9 CRSS 0.3
o Develop optimum schedule consistent with constraints of power, crew task selection, communications bandwidth, and non-interference among payloads and Space Station systems.	CRSS 7.2.1 CRSS 7.2.2.1 RFP[3] C-3-2.4g
o Revise schedule in accordance with changing requirements, priorities, opportunities, and capabilities.	CRSS 7.2.6
o Hold scheduled commands and dispatch at appropriate time.	CRSS 6.3.1.1
o Support onboard, near term planning, crew	RFP C-3-2.4c

[2] Customer Requirements for Standard Services, Reference 2

[3] Space Station RFP Section C, Reference 3

Table 1.3-4

Requirements for Resource Management

STEP	SOURCE
o Provide customers and operators data on Space Station resources and availability.	CRSS[2] 7.2.4
o Provide a single point of contact for Customer Communication Reallocation Requests.	CRSS 5.1.4
o Accommodate a phase degree of Space Station autonomy.	RFP[3] C-4-2.1.8
o Make Resource Management User Friendly.	CRSS 1.1.4 CRSS 1.1.8
o Ensure customer and core system do not interfere with each other and do not endanger the health and safety of the Space Station system.	CRSS 6.1.6.1

[2] Customer Requirements for Standard Services, Reference 2

[3] Space Station RFP Section C, Reference 3

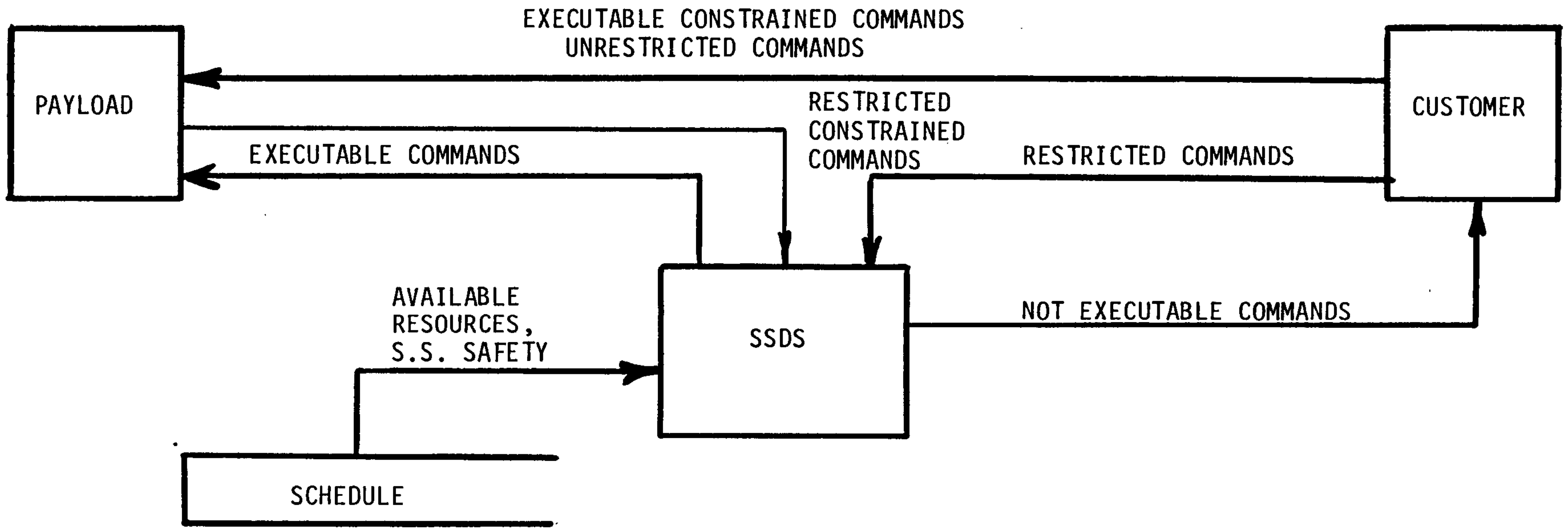


Figure 1.3-1. Command Management System Maximizing Customer Responsibility.

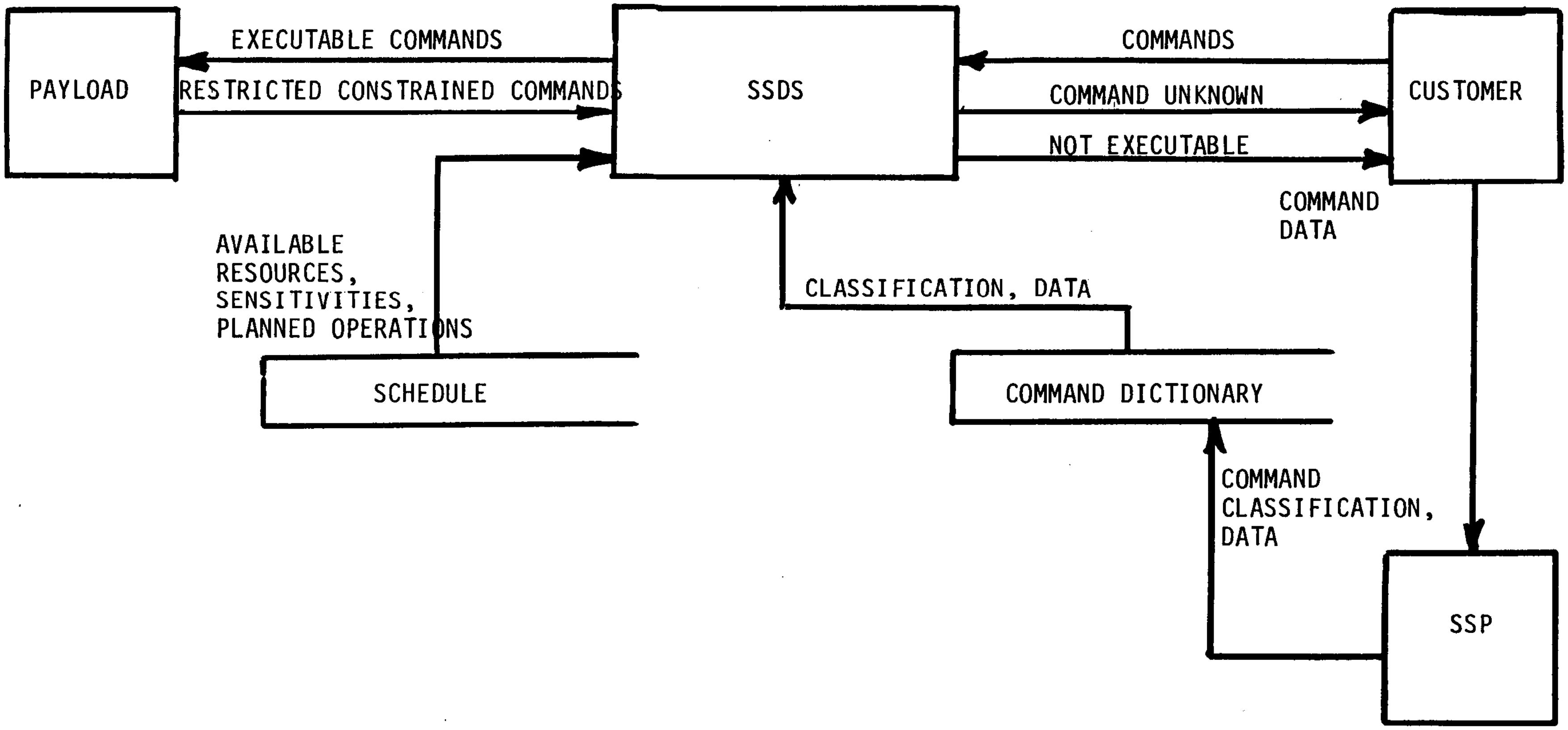


Figure 1.3-2. Command Management System Minimizing Customer Responsibility.

the customer. The SSDS may be required to read some or all commands to determine classification and executability, or some or all commands may have been previously classified and conditions for executability determined in order to allow the SSDS to process the command without sending it.

Since the purpose of this section is only to introduce the range of system options, implications and relative merits of these scenarios will not be discussed in this section. The more significant point is that a series of additions and function relocations can transform the system of Figure 1.3-1 to that of Figure 1.3-2. Hence, it is feasible to define a reference system which contains all of the steps of Table 1.3-1 and options which can transform it to either system extreme.

The reference system selected is shown in Figure 1.3-3. This system contains all of the essential steps from Table 1.3-1 and all of the elements necessary to represent the systems of Figures 1.3-1 and 1.3-2, as well as provisions for requirements from Table 1.3-2. The boundaries shown are those of the command management system, and do not represent any physical equipment or locations.

Commands may be issued by customers, operators, or their systems. The commands are authenticated. The valid commands are sorted by timing, with those for real time execution bypassing the command scheduling function. Those commands with no timing designation are treated as real time. This arrangement expedites the processing of interactive and other real time operating commands. It also passes non-restricted commands with no timing designation around the scheduling function.

Commands to be scheduled are passed to the schedule check function. Three possible determinations are made: scheduled, permissable, or need resolution. Scheduled commands are those for which prior provision has been made in the Space Station schedule. This provision will normally be as simple as scheduling a payload for operation. In some cases, specific provisions are required to draw on resources, vent, emit radiation, or perform other operations which may interfere with the operation of other payloads or core systems. Permissible commands are those not specifically provided in the

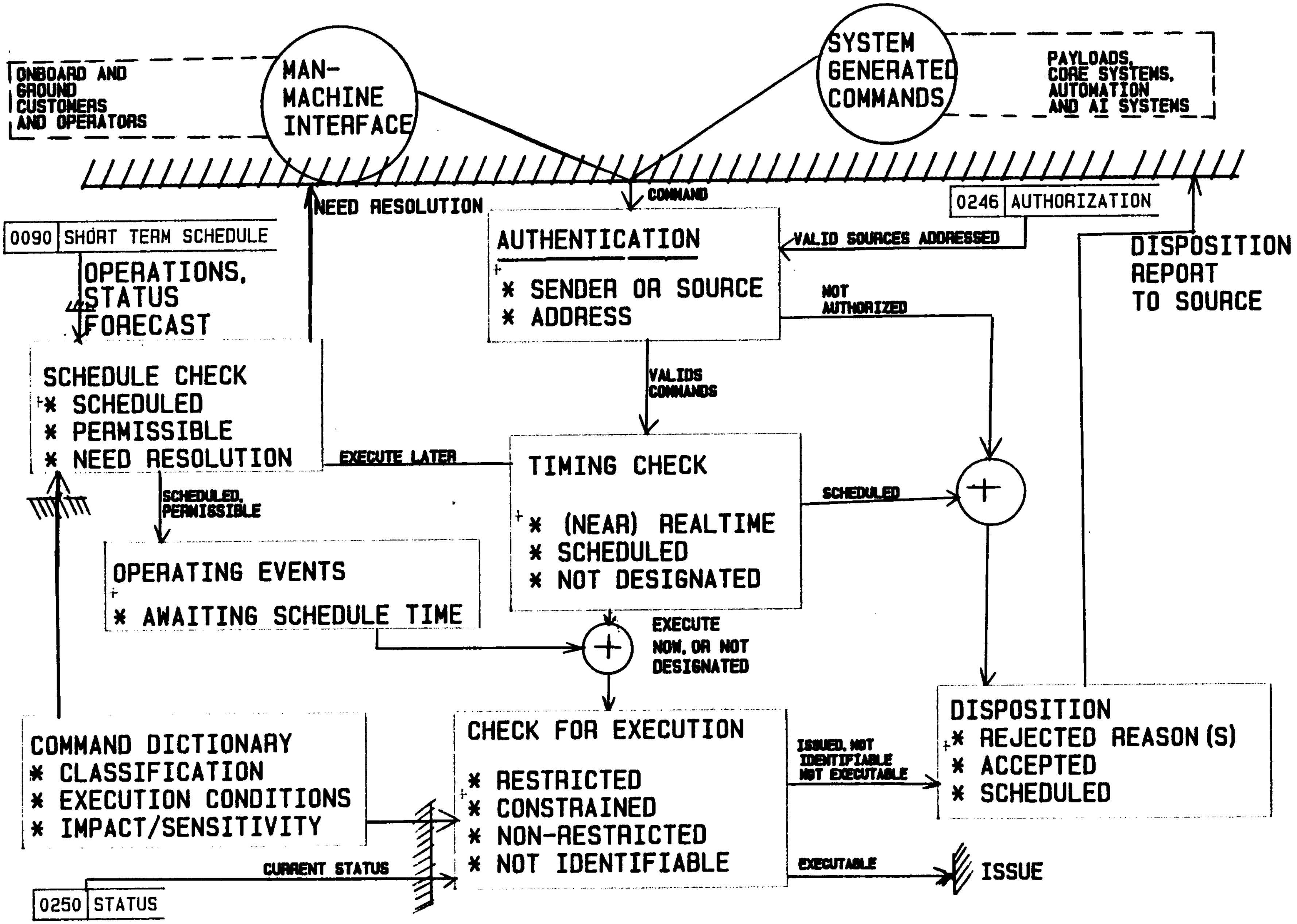


FIGURE 1.3-3 COMMAND MANAGEMENT REFERENCE SYSTEM FUNCTIONAL FLOW

schedule, but permissible under currently forecast conditions. Other commands require resolution to either modify the payload operations or the schedule to eliminate identified conflicts. These commands are referred to the customer and operator for resolution.

Scheduled and permissible commands are loaded into the operating schedule for execution at their requested times. A final check is made just prior to issuance to ensure that the commands are still executable. Those executable are issued, those not executable are reported back to their point of origin.

1.3.3 System Options and Reference System for Resource-Management

The two extremes of resource management are illustrated in Figures 1.3-4 and 1.3-5. Figure 1.3-4 depicts a system in which resources are allocated on demand according to whether a request for resources can be met. These requests may be issued by the customer or operator or automatically by their onboard or ground systems, e.g., payload processor. In this simplest implementation, requests that exceed capability are rejected.

The resource management system of Figure 1.3-5 utilizes a multi-tier scheduling process to allocate resources. A long term schedule is formed about a nucleus of program level events and the resultant master plan. The short term schedule adds detail and resolution to the long term schedule. Inputs may originate from customers, operators, or their automated systems. The operating schedule contains the individual, time phased commands necessary to implement the schedule. Additional levels of schedule could be added to reduce the size of steps between schedules.

Again, the intent of Figures 1.3-4 and 1.3-5 is to show extremes of attractive options and to provide guidance to the formulation of the reference system for resource management. This reference system must embody both extremes intrinsically, incorporate the essential steps for resource management from Table 1.3-3, and embody provisions for meeting the requirements of Table 1.3-4.

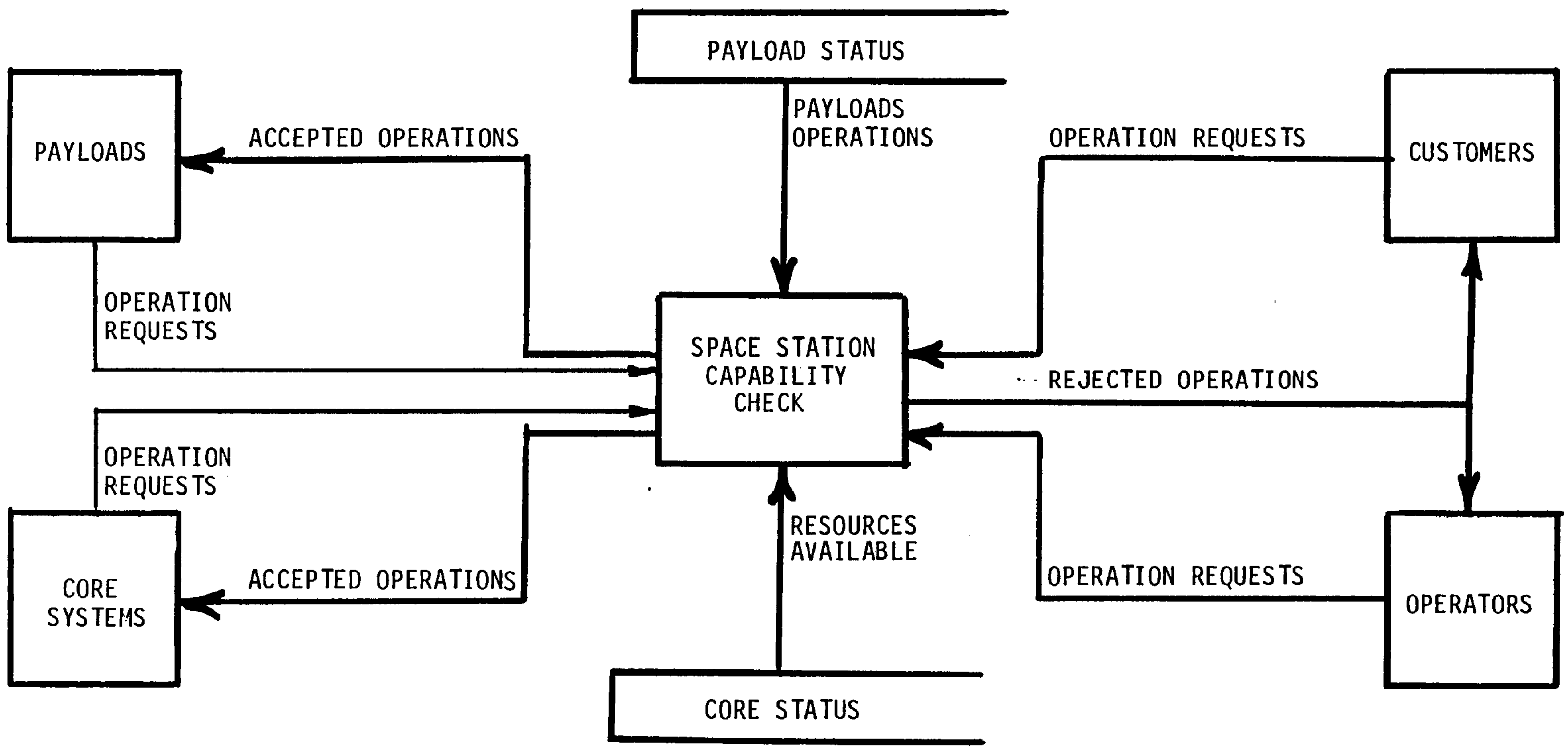


Figure 1.3-4. Resource Management by Operations Request.

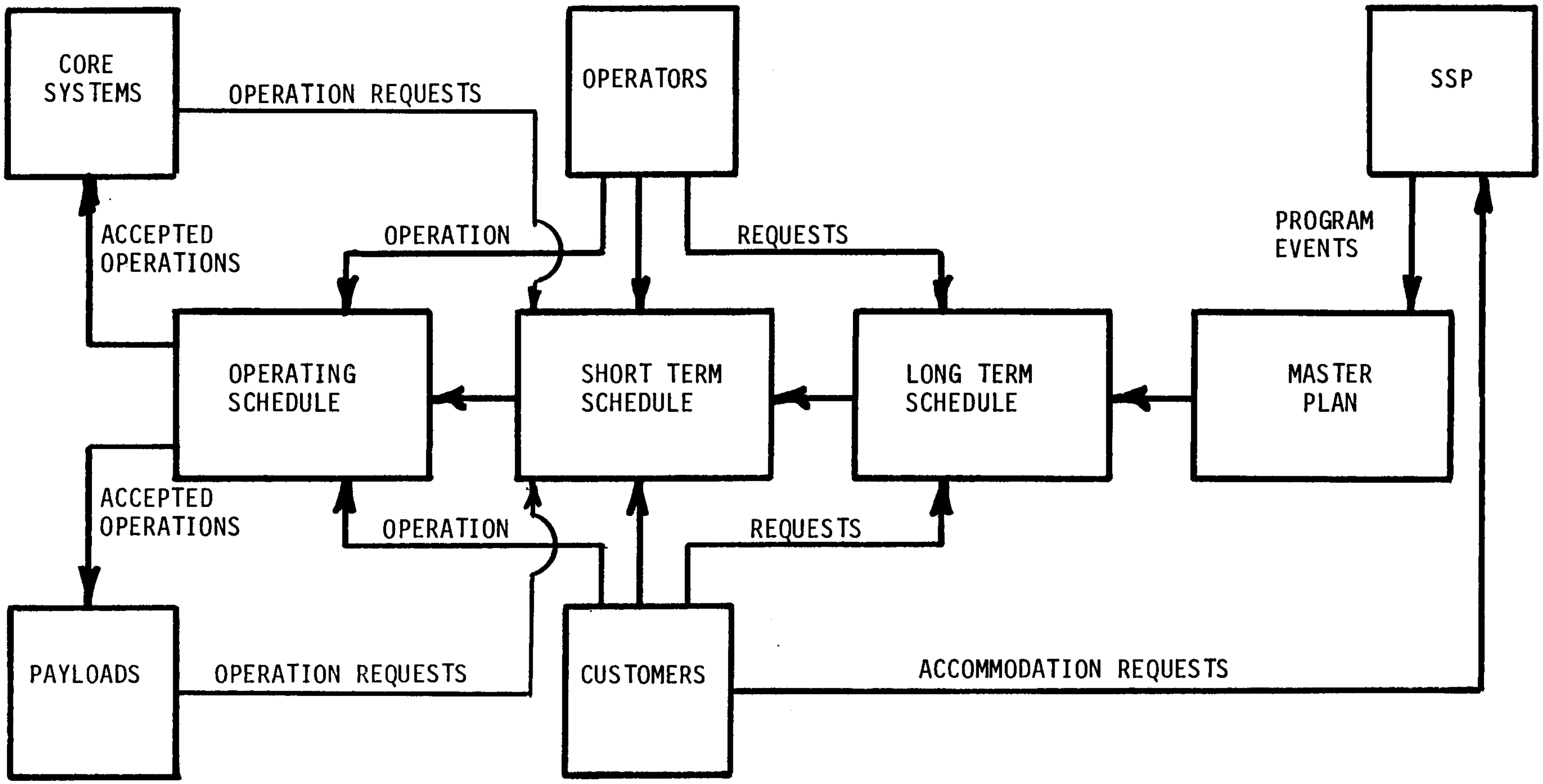


Figure 1.3-5. Resource Management by Multilayer Schedule.

The selected reference system is shown in Figure 1.3-6. Operations requests are received from customers, operators, and internal generation sources within their systems. The requests are validated and entered into the system. The requests are separated into those affecting existing schedules and those to be incorporated into new schedules. New schedule operations requests are accumulated until the time for the scheduling cycle to begin. A trial schedule is assembled and checked for conflicts such as resource limits, interferences, and Space Station safety. Conflicts are referred to the customers and operators for resolution. The schedule continues to be iterated to provide the progressive levels of detail required by the system. In addition, each level of schedule may be perturbed by new requests from the customers, operators, and their automated equipment, changes in resource forecasts (e.g., failed fuel cell reduces power available), major program event schedule changes, and priority changes. Onboard crew planning will influence the shortest term schedule, and may drive some aspects of that schedule.

The operating schedule is created from the shortest term schedule by the assembling of time tagged implementation commands entered by customers, operators and system equipment or stored within the system.

The number of levels of scheduling and the detailed content of each schedule is left unspecified in this reference system. Note that the case of resource allocation on demand bypasses the entire flow chart of Figure 1.3-6, as there is no schedule. However, the linking of command management and resource management through the operating schedule allows the execution check required by Figure 1.3-4 to be performed by command management, in the absence of schedules.

2.1.2.2.0 COMMAND MANAGEMENT

2.1 Description

Command management includes the entire end-to-end process of receiving customer and operator commands into the SSDS, validating and checking for executability, and delivering the commands to their addresses on the Space Station or Platform. Table 1.3-1 of Section 1.2.1 shows the steps essential

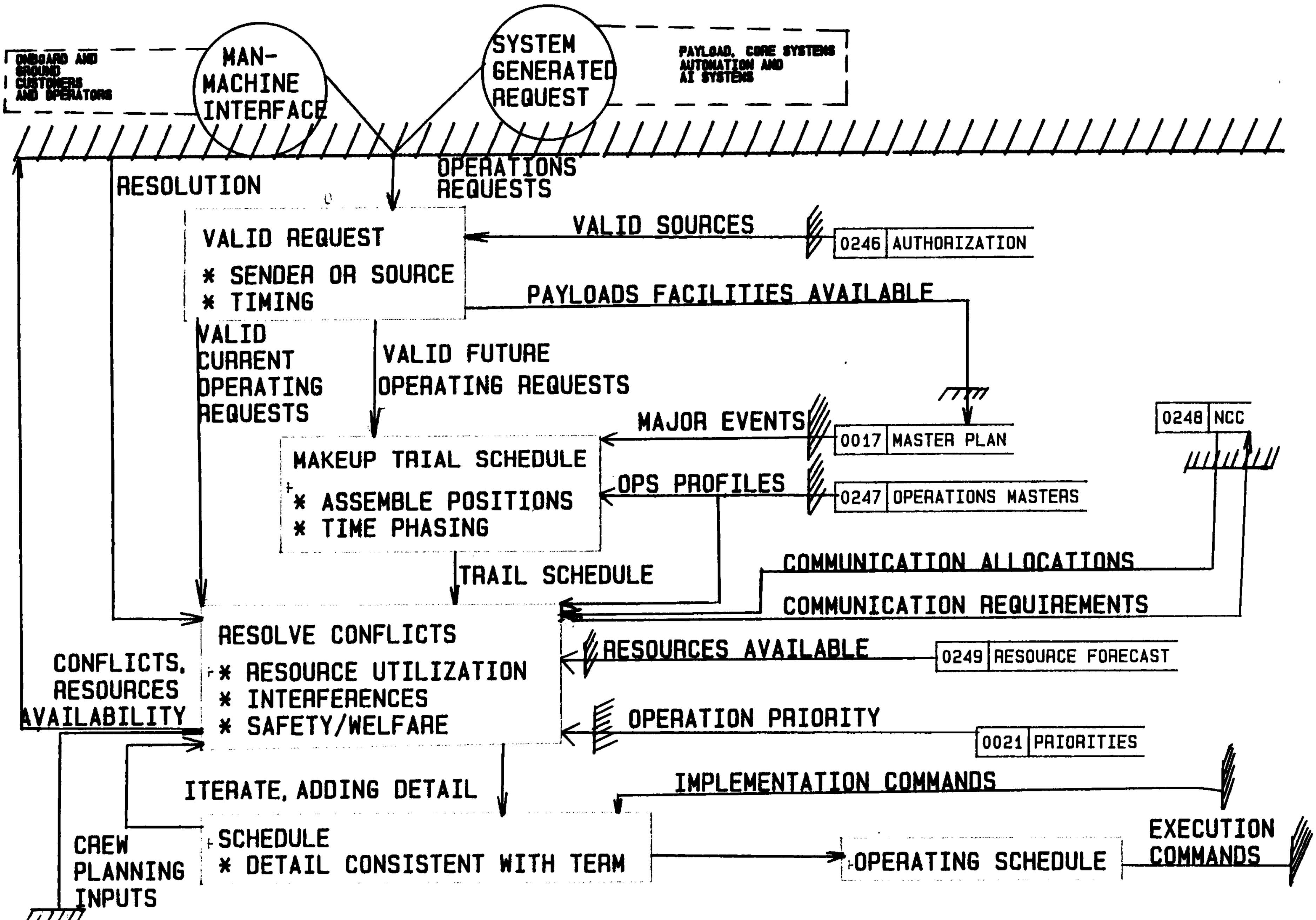


FIGURE 1.3-6 RESOURCE MANAGEMENT REFERENCE SYSTEM FUNCTIONAL FLOW

for command management. Variations in sequencing the given steps and routing of commands through the management process are to be considered in developing the command management function. Table 1.3-2 provides a list of representative command management functional requirements and references to detailed functional requirements provided in references 2 and 3. This section, command management, provides a brief description of "command" and indicates the command management functional steps selected for key options development and characterization.

2.1.1 Command

A command is a customer/operator user-defined message which is addressed to a specific application process onboard the receiving spacecraft [5]. For this discussion, a command will refer to the complete set of instructions necessary to implement an operation by the payload or subsystem being commanded. A command message is formatted with variable length so that the user/operator will not be constrained by the particular transport system.

A command can be a payload command or a core command issued by an authorized customer or operator. A core command is an operator command issued to a core system that maintains the environment in which the customer/operator performs his required services.

2.1.2 Customer/Operators

Customers are those people either on the ground or onboard, commanding payloads and receiving data by using Space Station Data System (SSDS) service. Operators are Space Station Program (SSP) personnel onboard and on the ground.

2.1.3 Payload

A payload is a customer supplied package mounted in or on the Space Station. Some payloads may utilize optional SSDS data processing and/or operator services.

2.2 Command Management Key Options and Options Characterization

The technical discussions given in previous sections, cover briefly the command definition, the customer/operator definition, the essential command management functional steps and requirements, and the functional interface processing between command management and resource management. These steps of command management selected for further technical discussions and options development are:

- Command Format
- Command Classification
- Checking for Command Executability
- Priority

Each of these steps will be described in detail. Options will be discussed based on the description. The selection of the option to be implemented will be left to the Task 3 trade studies or the Task 4 design activity.

2.2.1 Command Format

2.2.1.1 Description

a. Command Recognition. As specified in section 5.3.2.1 of reference 1, upon receipt of a user command, the customer/operator identification shall be checked to ensure that the customer/operator is authorized to issue the command or data to the address in the header. If unauthorized, the command will be rejected, and the originator notified.

b. CCSDS Command Formatting. As shown in [5], the CCSDS (Consultative Committee for Space Data System) has designed a standardized telecommand packet (TC packet) for a customer/operator command containing a user-defined message which is addressed to a specific application process onboard the receiving spacecraft. A TC packet, formatted for a user-command, is the data entry passed end-to-end from the point of origin (e.g., the customer's facility) to spacecraft (or spacecraft to spacecraft) data network. Figure 2.2-1 illustrates the telecommand packet format as proposed by CCSDS [5].

The primary header consists of three data fields: the packet identification (including application process ID and version number), packet sequence control (including packet name or sequence count), and packet length between the first bit of the secondary header and the last bit of the optional packet error control. The optional secondary header is designed to provide ancillary data for interpretation of the command data in the packet. The command data field contains the user command information to be used by the application process. For very long packets device of packet segmentation and use of virtual channel are proposed by CCSDS for implementation for uplink telecommand transmission [6].

c. CCSDS Command Transmission from Sending to Receiving. The customer/operator's command, formatted as a telecommand packet, is submitted to the SSDS for transmission. It can be a simple TC packet, a sequence of packets, or a file of packets, together with command data information concerning operation procedures to be used. Figure 2.2-2 illustrates the logic flow of the CCSDS proposed layered command formatting for end-to-end command handling and data network processing [5]. A user application process presents user packet(s) of information from the application process layer to interface with the packet layer. The packet layer may append packet(s) containing transfer control directions in accordance with transfer service requests. The transfer frame layer service is provided by the sender to the receiver jointly performing agreed command operation procedures within transfer frame layers [6]. The services of the TC channel coding layer, which provides a standard encoding technique for required reliability of bit transfer, are used to establish and operate the physical data channel [7].

The receiving transfer frame layer is responsible for transfering the transfer frames reliably to the receiving packet layer onboard the receiving spacecraft, where the packet assembly process in the packet layer reassembles each packet and monitors its delivery to the designated receiving application process.

This completes the CCSDS proposed layered command data network logic from ground to space for uplink telecommand transmission.

According to the command data information from a sending packet addressed to the receiving application process, mission data are generated, assembled, and delivered to the ground user via telemetry packets as defined in the CCSDS blue books [8] [9].

d. User Control Requirements for Command Processing.

o User Privacy Requirement: As shown in [2], users requiring secure commanding of their payloads shall be responsible for command encryption and decryption within the payload and on ground. Therefore, the SSDS should be able to support user's payload operation without reading the user;s command.

o Real Time Data for Interactive Control: As specified in [1] and [2], the SSDS shall provide the user a real time, interactive control capability for non-restricted commands and shall include restricted and constrained commands, in so far as possible.

o Time Tagged Commands: All restricted and constrained commands must be time tagged options for time tagged non-restricted commands.

2.2.1.2 Option Characterization

The command format option characterization will be discussed in the order of the command format descriptions given in the previous section 2.2.1.1, as follows:

a. User Identification and Command Recognition

o The customer/operator - The user command for payloads can be generated at or issued from at least the following three locations [2]: the user's facility, the ground control center(s), and the Space Station. The SSDS must have a way to determine whether the user is authorized to command that payload.

Some user or system control requirement, such as a data base user name directory, must be provided for checking user identification and authorized commands. This user ID is also important for end-to-end data distribution (after demultiplexing of transfer frames at the ground data processing control centers) to return mission data to the user who had issued the command. There is no mention of the user ID and authorization in the development of the telecommand packet standards [5].

The customer/operator identification can be implemented easily by any of several methods in current commercial practice. The merits of some alternative approaches will be considered in Task 4. The major concern is with any impact SSDS requirement may have on the development of the CCSDS/ISO standards for end-to-end data interface networking.

- Command Recognition - Each command must be classified as non-restricted, constrained, or restricted. Command processing will depend on this classification. Since some customers will want to maintain command security, it is desirable to process commands based on information in the command headers.

  This can be done by a command identification code, by only allowing encryption of non-restricted commands, or by customer determination of which commands may be executed.

b. Use of CCSDS Standards and the ISO/OSI Reference Model for Telecommand Transmission. As defined in [2], user interface standards shall be defined in accordance with the International Standards Organization (ISO) 7-layer reference model for open system interconnect (OSI) as first preference. The SSDS reference system (as it is defined in previous section 1.3) shall use each of the 7-layers existing internationally accepted standards at first priority. When practical, appropriate standards from the sources such as the CCSDS layered design standards as shown in Figure 2.2-2, shall be used at the upper layers to supplement the ISO/OSI 7-layer reference model.

d. User Control Requirements for Real-Time Operation and Data Distribution

1. Option characterization of User Requirements for Real-Time Operation - As specified in section 6.2 in [2], the SSDS reference system shall:

o Support user operation of their payloads in real-time.

o Permit the use of customer unique software to support the real-time interaction which applies to non-restricted commands, and to restricted and constrained commands to the maximum extent possible.

o Make the real-time interaction capability available to the flight crew or the ground users.

o Support the ground or payload specialist to change automated command sequences in real-time operation mode.

o These user requirements included in the command for real-time operation can be added to the CCSDS/ISO primary or secondary header. The standards or format requirement specifications for the user real-time operation will be left to Task 3 for trade studies and to Task 4 for system design.

2. Option characterization of User Requirements for Real-Time Data Distribution - The SSDS reference system shall provide users with real-time and near-real-time data return by performing the following:

o Provide real-time payload data.

o Provide level-0 payload and monitor data in near-real-time for quick-look monitoring.

o Deliver core system monitor data for ground operators in real-time or near-real-time.

- Account for real-time and near-real-time data until all data are delivered.

- Provide real-time distribution of real-time and near-real-time data, including level-0 processing, demultiplexing, buffering, routing, and retransmission in accordance to user telecommand.

- These user requirements for level-0 payload real-time and quick look data distribution should be contained in the user command headers (primary and secondary), to enable the SSDS to arrange communications links and bandwidth to support the end-to-end networking.

- The user command should contain a tagged timing requirement parameter: real-time or near-real-time defined as zero hour timing, or a specified time for the command to be issued.

2.2.2 Command Classification

2.2.2.1 Description

As described in previous section 1.3, the customer/operator command shall be checked to see whether the command is issued by an authorized user. If so, the command shall be checked against the command dictionary for authentication and then differentiated for identification. Consequently, the user command shall be handled appropriately based on its classification.

2.2.2.2 Option Characterization of Command Classification

a. General Option Discussions. The option of command classification shall be characterized, in general, by following the technical discussions given in section 4.3.4 in [1], as follows.

- The ground user command shall be verified, checked, and classified prior to telecommand delivery to the payload.

- The customer/operator command issued onboard shall be verified, checked, and classified onboard the space station.

- The classification of an authorized user command shall be determined (against the data base command dictionary) in the following three (3) types:
  - Non-restricted commands, which are neither restricted nor constrained.
  - Restricted commands, which are payload/core commands that could endanger the SS safety or its payload and constellation elements.
  - Constrained commands, which are those payloads that require coordination with other users or core systems to insure that users do not interfere with each other and to coordinate SS systems support of customer requests for services.

- According to its executability, a restricted or constrained, i.e. command can be determined as executable cleared for execution on a specific schedule.

- Non-restricted commands and data are passed directly to payload or core with no further checking beyond the address of the destination.

- Restricted and constrained commands receive command management with conditions checking for command executability by one of several alternative processes.

- Some restricted and constrained may be processed for immediate execution, consistent with real time, interactive control. They will be routed directly to their destination without further delay. The goal is no significant increase in delay over that for non-restricted commands.

- Some restricted and constrained commands may be processed for later execution. The command accepted for later execution may become unexecutable prior to its scheduled time due to unforeseeable system or payload status changes.

- Commands executable at a scheduled time are held until the time for its release, then routed to the payload for execution.

- Some commands being held for later execution may become not executable due to the schedule changes resulting from equipment failure, system capability loss, or preemption by other commands with higher priorities. The commands so affected will be treated as any other "not executable" command.

- Commands which are not executable will be referred to the customer/operator for re-planning and re-scheduling.

b. Specific Option Discussions. Other options of command classification that are noteworthy are briefly discussed as follows.

- The command identification process is completely apriori. That implies the command classification is determined apriori (pre-determined).

- Another concept is that the command classification is accomplished at the time it is submitted.

- As described in previous section 1.3.2, in one extreme of command management, customer takes the responsibility to classify commands as restricted, constrained, or unconstrained. Customer determines the conditions for executability of constrained command. Only non-restricted and executable, constrained commands are passed to the payloads. The SSDS has only the limited responsibility of determining the executability of restricted commands.

- In another command management extremity, as shown in Figure 1.3-2, customer plays only limited role in command classification. The SSDS receives and processes all customer/operator commands. By checking against the data base command dictionary, the executability of the command is determined. Unknown (unlisted) commands issued by authorized customers will be rejected.

The requirements specification for the command classification, whether it is apriori, as submitted, customer determination, or SSDS-does-it all, will be left to Task 3 for trade studies and to Task 4 for system design.

2.2.3 Checking for Command Executability

2.2.3.1 Description

As briefly described in section 1.3.2, there are two extremities in the command management system for checking the command executability. One extreme of the reference system claims full responsibility for SSDS to read all customer commands, to classify them, and to check their executability. The other extremity places maximum responsibility to the customer for determining command classification, checking executability of constrained commands, and sending commands to the payloads or SSDS, as appropriate.

At the same time, each of the extremities will interact with the resource management for command scheduling by checking against the resources availability and limitations.

The SSDS command management system shall be responsive during checking for command executability to support alerts and rapid replanning for unique payload opportunities (such as volcano eruptions and solar flares) in a real-time or near real-time operation mode.

2.2.3.2 Option Characterization

The option characterization of checking for command executability will be discussed as follows.

a. All customer responsibility - One option is to designate the customer with full responsibility to check constrained command executability. In a command management system adopting this extreme approach, the customer shall be required to:

   o Issue all commands.

- Determine all command classifications as to non-restricted, constrained, or restricted command.

- Check constrained command executability and send the non-restricted and executable, constrained commands to the payload.

- Send only restricted commands to SSDS which, in turn, interacts with the resource management system by checking against resources availability, safety and other limitations. If the commands are executable, they are sent to the payload. Otherwise, SSDS wil send the unexecutable commands back to the customer and provide assistance in clearing for execution, to include at least the reasons for non-executability.

  This extreme minimizes SSDS's responsibility to process only restricted user commands.

b. All SSDS Responsibility - Another extreme in the command management reference system places minimal responsibility on customer. SSDS has full responsibility in checking for command executability by performing the following:

- Check the timing tag included in the user command as to whether it is real-time, or near-real-time, scheduled, or undesignated.

- Check for executability with: (1) information data provided by command dictionary: command classifications, conditions for command executions, sensitivity and impact associated with command payload operations, and (2) information data provided by resource management: resources availability and limitations, sensitivities and scheduled/planned operations, and (3) command priorities.

- The executable, constrained/restricted commands and non-restricted commands will be sent to payload for execution. Both will be consistent with real-time operations, in so far as practical.

- The not identifiable and not executable commands, together with command status will be sent to the disposition function in the command management. Appropriate assistance will be made available to the customer to aid in resolution of not-executable commands.

c. Intermediate Options - Between these extremes lie many optional systems which will resolve difficulties with the extremes and maximize benefits. These options will all be considered in the Task 3 trade studies.

The implementation of Checking for Command Executability will be left to Task 3 for trade studies and to Task 4 for system design.

2.2.4 Priority

2.2.4.1 Description

As specified in Section 5.1.4 in [2], the SSDS reference system shall support the capability for on demand reallocation of data resources to meet customer priorities. In the extreme application, customers with highest priority will get what they have requested all time, provided the requested resources are available and sufficient. Customers with lower (or lowest) priorities will be scheduled later (or last) or never, due to limited resources. This options area is concerned with the various ways of utilizing priorities, short of this extreme application.

Priorities can be established by the policy-making space station program committee or through negotiations among customers and the committee. As stated in Section 5.1.2 in [2], SSDS shall manage transfer of customer data. SSDS determines the priority of the customers' requests for data transfer based on customer data accessibility, data volume. Then SSDS shall assign valid custmer data transfer requests to demand or batch priority.

Under certain circumstances SSDS shall accommodate customer requests for dynamic priorities by interrupting some operation(s) in current execution and executing the customer requested operation(s) on demand a in real-time operation mode.

2.2.4.2 Option Characterization

Option characterization of customer priority implementation will be discussed as follows.

a. Straight Priority Versus Time Allocation

- Straight priority can be simply defined in the case of command management, as an orderly command processing, such that the command, which carry a higher priority request will be processed and implemented in preference to commands carrying a lower priority.

- Priority is usually assigned by customer. In some cases, when customers choose to ignore, SSDS will have to assign priorities, such as FIFO (first in, first out), and LIFO (last in first out). These are often referred to as queueing disciplines in queueing theory.

- Time allocation for operation is an alternative for priority implementation. Customers with high priority are allocated more operating time (or a higher proportion of critical resources), but all customers will be guaranteed an operating time commensurate with priority and mission objectives.

b. Charge for Priority Versus Allocation by Need or Merit

- Priority assignment is usually associated with a fee. That is, a customer command assigned with a higher priority will pay more for the services supplied by the SSP.

- The criterion for assigning a high or low priority to a customer may absolve based on some measure of need or merit. The factors given below should be considered as a basis for priority determination:

  1. Value Judgment
  2. Fee Structure
  3. Pecking Order
  4. Political Considerations
  5. Impacts of Resource Demands

c. Dynamic Priority for Unique Payload Opportunities

- o Dynamic priorities often arise under unexpected circumstances, such as volcano eruption and payload observation over a specific ground target under specific circumstances. This kind of dynamic priority request for unique payload opportunities may be accommodated by the SSDS resource management system by flexible scheduling algorithms. However, the development cost and complexity grows with the degree of flexibility provided..

The implementation of managing user command priorities will be left to Task 3 for trade studies and to Task 4 for system design.

2.1.2.3.0 RESOURCE MANAGEMENT

3.1 Description

A brief discussion on system options and reference system for resource management was given in previous Section 1.3.3. This section, in addition, provides some generalized descriptions on resource management as follows.

- o Overall SSDS Scheduling Requirements
  The SSDS overall scheduling requirement is to provide for short term scheduling and executing of all operations required by the space station customers and operators. The SSDS shall support customer/operator interactive development of appropriate operating schedules based on master plan data developed by the Space Station Program (SSP).

- o The SSDS Scheduling Objective
  The SSDS develops an "optimum" schedule which integrates multiple customer requests and payload accommodation requirements consistent with constraints of Space Station resources.

- Critical Space Station Resources and Constraints

  The critical Space Station resources and constraints include:

  - Power and Electrical Energy
  - Crew Activity and Flexibility to Accommodate Crew Task Selection
  - Uplink/Downlink Bandwidth
  - Crew and Space Station or Platform Safety
  - Payload and Space Station or Platform System Interaction and Interference
  - Data processing resources that include computers, storage devices, and networks supporting the SSDS data processing facilities.

- The Master Plan

  The term "Master Plan" requires definition in order to understand the envisioned interface between the Space Station Program and the SSDS resource management function. The Master Plan is a listing of the major program level events that will affect the scheduling of Space Station and payload operations. A partial list of these events is given in Table 3.1-1, together with data sources and responsibilities. The data required in the Master Plan are generally appropriate for collection and coordination by the Space Station Program. The Master Plan will be maintained by the SSDS.

Table 3.1–1 Major Program Events Included in the Master Plan

EVENT	SOURCE–RESPONSIBILITY
o NSTS Orbiter Visits – Date – Duration – Cargo Manifest – Return Cargo	SSP – Planning, Coordination with MCC – Initial Schedule MCC – Schedule Updates
o Payload Changes, Servicing – Additions – Returns – Upgrade/Refurbish	SSP – Coordinate Requirements with Customers – Coordinate Transportation with MCC
o OTV/OMV Flights – Date – Mission, Payload – Duration	SSP – Coordinate with Customers – Initial Schedule – Schedule Updates

Table 3.1-1 Major Program Events Included in the Master Plan (Cont'd)

EVENT	SOURCE-RESPONSIBILITY
o Free Flier Servicing - Date - Free Flier - Sensitivities, Interferences - Time Line	SSP - Coordinate Schedule with Customer - Identify Restrictions, Constraints Customer - Provide Mission Data
o SSPE Servicing - Date - Element - Sensitivities, Interferences - Time Line	SSP - Coordinate Schedule - Identify Restrictions, Constraints SSPE - Provide Mission Data
o Space Station Additions - Date - Resource Impact - Addition Time Line - Sensitivities, Interferences	SSP - Coordinate Transportation with MCC - Provide Resource Impact, i.e. Capabilities, Demand - Identify Changes in Reference Data, e.g. Restrictions, Constraints, Priorities - Provide Mission Impact Data

3.2 Resource Management Options and Option Characterization

In the area of resource management, selected for further technical discussions are the following three (3) key options:

- o Schedule objectives
- o SSDS scheduling scope
- o Scheduling conflicts resolution

Each of the key options will be described and option characterization will be discussed based on the option description. The option resolutions and implementations will be left to Task 3 for trade studies and to Task 4 for system design.

3.2.1 Schedule Objectives

3.2.1.1 Description

The scheduling function in the resource management system provides the Space Station operating schedules based on customer/operator requests which demand Space Station resources. The SSDS scheduling can be accomplished: a) interactively in most cases with the customers/operators who issued the requests and commands, b) automatically with the aid of some automated scheduling algorithms, such as the "expert system" derived from artificial intelligence which will be briefly discussed in the next section for option characterization, and c) manually - sometimes it is still considered as practical and operational. The major objective of the scheduling function in the SSDS resource management is to develop optimum schedules which are consistent with the constraints of power, crew time, communication link bandwidth, and other SSDS resources within limits that ensure that customer and core systems do not interfere with each other and do not endanger the health and safety of the space station system.

Optimization must be performed on some quantitative figure of merit, such as the value of products generated by the payloads or the adequate allocation of Space Station resources and services to a spectrum of customers. Other figures of merits can be summarized as follows:

- Accountability for success and loss of operations.
- Accommodation of all customers (including customer commands with low priorities).
- Maximum utilization of all resources, especially critical resources.

3.2.1.2 Option Characterization

The option characterization of schedule objectives is discussed in the following:

- Maximum Resource Utilization versus Value of Products: The utilization of the Space Station resources, such as data processing throughput, buffer allocation, power/energy communications bandwidth, and crew time can be optimized with existing conventional linear/non-linear programming algorithms, so that the value of products is optimized in accordance with customer requirements, priorities, and opportunities. However, if the Space Station resource utilization is maximized in order to optimize the value of products, it will be more difficult to accommodate customer dynamic priority demands for resources and services.

- Flexibility versus Optimization: the primary objective of the SSDS scheduling function can be optimized between the use of the Space Station resources to maximize the value of results to the customers and provide a flexible schedule which is able to automatically or interactively accommodate changing customer requirements, priorities, and opportunities.

- Priorities versus Accommodation of all customer's objectives: The arguments presented in the above two paragraphs can be equally applied here.

- Expert System (using artificial intelligence) for scheduling: A few schedulers have already been developed in the aerospace sector by using expert system approaches and several others are currently under development. Dr. Karl Kempf, Section Manager of the newly formed Artificial Intelligence Laboratory at MDRL, is specializing in software for control of automated production systems involving robotics and scheduling with expert systems. MDTSCO at Houston and KSC, both are under contract with NASA to develop expert schedulers for two different appliations. The KSC application is for expert scheduling space vehicle fueling sequences of events, and the other at Houston is for an expert scheduler for scheduling launch and flight operations. It should be noted that, in general, expert schedulers such as those mentioned above are not necessarily reconfigurable to a new application, such as an expert SSDS scheduler.

Expert schedulers are in the current state-of-art of technology and in use in several applications. They are not, however, an off-the-shelf item. The development of expert system technology tailored for an expert SSDS scheduler requires Tasks 3 and 4 activities.

3.2.2 SSDS Scheduling Scope

3.2.2.1 Description

As specified in [1], SSDS plays a major role in scheduling. The Space Station Program develops the master plan impacts for program schedules as discussed in detail in previous section 3.1. The customers/operators on the ground and crew on board participate in schedule development and scheduling conflicts resolutions.

3.2.2.2 Option Characterization

This section provides brief technical discussions on the option characterization of the SSDS scheduling scope as follows.

a. Long Term Schedule or No Schedule: No Schedule does not mean there will be no SSDS scheduling at all. A no-schedule approach is accomplished by simply performing the following:

- Receive the customer/operator requests demanding SSDS resources.

- Check customer/operator requests against the Space Station resource availability and command restrictions and constraints.

- If Space Station resources are available and meet customer demands, and the customer/operation command is otherwise executable, the command will be accepted and delivered for execution.

- Otherwise, customer/operator command will be rejected.

Additional levels of schedule which may be provided to aid in optimizing the schedule and meeting customer and program requirements include:

1) Operating Events Schedule - A short to very short term schedule containing time tagged commands to implement events planned for operation.

2) Short Term Schedule - Used to optimize and hold the events planned for operation. Would normally employ interactive optimization by both customers and system operators.

3) Intermediate Term Schedule - Provides information similar to the Short Term Schedule, but with less resolution.

4) Long Term Schedule - Provides estimated block allocations of operating time at low resolution.

5) Master Plan - Contains program events which will be major perturbations to "normal" operations.

The role and value of these levels of schedule will be assessed in the Task 3 and 4 activities.

b. Responsibility for SSDS scheduling to meet customer objectives: As specified in sections 5.3.3 and 6.2 in [1], the resources management reference system develops the following schedules to meet customer objectives:

1) Develop master plan as discussed in previous section 3.1.

2) Retain and maintain normal day and major event day schedules. These schedules may be used as baselines for developing schedules for normal and major event days, by

- Supporting interactive payload planning and scheduling.

- Integrating customer payload operating requests with Space Station resource availability.

- Supporting customer requests for non-SSIS resources.

- Providing projected data on Space Station resources.

- Interactively developing integrated Space Station element schedules.

3) Develop short term schedules: The SSDS shall support the interactive development of short term Space Station schedules by Space Station system operators and customers. Specifically, the SSDS shall:

- Allow near-term planning entries by the onboard crew.

- Coordinate communications allocations for all SSPE's with the network control center. Resolve all SSP requirements conflicts. Provide type of service, data rates, desired times, and allocations to SSPE's.

- Support interactive planning by payload and core system operators including significant orbital parameters, such as time, orbital position, and object ephemerides, which may influence payload and Space Station operations.

- Provide indications of Space Station resource capabilities and availability to support payload output optimization.

- Optimize utilization of Space Station resources and maximization of payload output.

- Support rapid replanning to allow customers to take advantage of payload opportunities.

- Support clearing of restricted and constrained payload commands to assist in real-time payload operation.

- Support customer requests for non-SSIS resources.

- Develop integrated SSP operations schedules.

- Provide accountability of the disposition of restricted and constrained commands.

- Schedule use of Space Station resources by other SSPE's (e.g., platform - space station communications, crew time, etc.) and resolve conflicting requests.

- Support interactive schedule development with appropriate data processing input/output and display.

4) Develop operating events schedules: The SSDS shall develop an operating event schedule containing the time sequenced operating events in the short term schedule, together with the commands necessary to initiate these operations. Specifically, the SSDS shall:

   - Check restricted commands and constrained commands for execution within the existing short term schedule.

   - Deliver immediately or incorporate into the operating event schedule, as appropriate, those restricted and constrained payload and core commands which are executable.

   - Alert the customer and operator to those restricted and constrained payload commands which are not executable in the current short terms schedule. Transfer these commands to Function 3.2 for disposition.

- Support real-time reallocation of data distribution resources to help meet customer priorities. Coordinate communications requirements with network control.

- Store customer payload commands for execution according to a schedule, sequence of events, or another stored command.

- Provide accounting of commands incorporated into the operating event schedule.

3.2.3 Scheduling Conflicts Resolution

3.2.3.1 Description

The final area for option development in resource management is resolution of customer commands which exceed constraints or violate restrictions. The scheduling conflict conditions may result from the following circumstances:

- Scheduling which takes place closer to the time of execution frequently involves conflicts resolution between customers requests for limited Space Station resources.

- Customers attempting to operate their equipment in a manner not anticipated in the schedule.

- Space Station equipment failure or other loss of capability.

- Schedule preemption by a user higher priority operation.

- Error conditions.

- Operations interference between customers.

3.2.3.2 Option Characterization

This section provides technical discussions on the option charcterization of the scheduling conflicts resolution as follows:

a. Utilization Level of Resources: The utilization level of Space Station resources has to be optimized in such a way that optimum schedules can be developed consistently with constraints of power, crew activity, communications bandwidth, and non-interference among payloads and space station systems. Desired level of resource utilization can be established by using mathematical optimization techniques or expert system technologies through simulation modeling.

b. Allowance for Opportunities: To avoid scheduling conflicts resolution, SSDS shall maintain resource utilization at an appropriate level and support short term schedule revisions down to the last minute for execution to accommodate customer dynamic changes in payload observations and target of opportunity operations. At the same time, customer desires interactive involvement with SSDS in system schedule development, including real-time adjustments to accommodate target of opportunity without causing any scheduling conflict.

c. Early versus "On The Fly" Resource Management:

1) Early resource management includes:

- o Development of major event schedule (master plan) as discussed in section 3.1.

- o Development of single or multiple layered schedules, and operating event schedules as discussed in previous section 3.2.2.2.

- o Short term schedule revision close to the time of execution to accommodate customer changes in payload observations.

- o Real-time scheduling adjustment to accommodate customer targets of opportunity.

- o Maintenance of an adequate resource utilization level for optimum scheduling without causing any scheduling conflicts.

2) "On the fly" resource management allocates space station resources as they are going to be utilized to fulfill customer requests. If customer commands can be scheduled with available resources at a time satisfactory to the customer, it will be accepted and scheduled for execution. Otherwise, customer commands must be rejected.

3) Scheduling conflicts resolution:

- o The SSDS should attempt to resolve scheduling conflicts.

- o Resolution can be resolved by an operator, automatically by the SSDS software, or interactively, with or without customer/operator involvement.

- o An interactive procedure for conflicts resolution with customer/operator involvement is currently preferred.

- o In case scheduling conflicts cannot be resolved, customers must be informed of the reasons for rejection, because they may need assistance in replanning. In this case, the SSDS will be required to provide reasons for rejection, and the Space Station program provides assistance in replanning.

2.1.2.4.0 TRACEABILITY CROSS REFERENCE MATRIX OF OPTIONS VS RFP PARAGRAPH NUMBERS (AND REQ # DESCRIPTOR)

The cross reference matrix is given, by definition, for cross referencing the technical discussions on selected options with their associated RFP requirements. The top-down and bottom-up traceability matrices are presented in Table I Appendix A-5 and A-6 respectively.

In column 1, the number in the form of 2.1, e.g., indicates paragraph 2.1 at level 3 in section 5 of [1]. Whereas paragraph 2.0 relates "Command Management", and paragraph 3.0 relates "Resource Management", in this paper.

The number in column 4 in the form of 6.2.1.2, e.g., indicates paragraph 6.2.1.2 of [2], and c-3-2.4.g indicates paragraph 2.4.g on page c-3 of [3].

2.1.2. 5.0 REFERENCES

[1] "Space Station Data System Analysis/Architecture Study, Task 1 – Functional Requirements Definition", McDonnell Douglas Astronautics Company Report MDC H1343, revision 1, dated January 11, 1985.

[2] "Customer Requirements for Standard Services from the Space Station Information System (SSIS)", Revision 1 NASA GSFC Space Station Office, dated September 19, 1984.

[3] "Space Station Definition and Preliminary Design RFP", NASA, dated September 15, 1984.

[4] Space Station Mission Data Base, Woods Hole Mission Requirements Working Group, draft, undated.

[5] "Telecommand, Part-3: Packet Formats and Procedures," White Book, Issue-0, CCSDS, April 1984.

[6] "Telecommand, Part-2: Transfer Frame Formats and Procedures," White Book, Issue-3, CCSDS, July 1984.

[7] "Telecommand, Part-1: Channel Coding and Procedures," Red Book, Issue-0, CCSDS, May 1984.

[8] "Packet Telemetry," Blue Book, CCSDS, May 1984.

[9] "Telemetry Channel Coding," Blue Book, CCSDS, May 1984.

[10] "Working Group Meeting On the Development of One Option for the Use of CCSDS and the ISO/OSI Reference Model, and Their Associated Specifications, for Use in the SSDS," by J. J. Zapalac, MDAC, 26 February 1985.

[11] J. J. Lee and K. Y. Liu, "Recent Results on the Use of Concatenated Reed-Solomon/Viterbi Channel Coding and Data Compression for Space Communications," IEEE Transactions on Communications, Vol. COM-32, No. 5, May 1984.

[12] "An Introduction to PAWS - Performance Analysts Workbench System," Information Research Associates, Austin, Texas, October 1982.

2.1.3 DISTRIBUTED OPERATING SYSTEM DESIGN

This white paper supports the trade study of operating systems for the Space Station project.

What is an operating system? An operating system (OS) can be roughly thought of as the interface between a user program and the hardware on which it is meant to run, although some definitions also include library facilities such as data format conversion routines. Traditionally, this interface has involved the distribution of limited resources among a number of users, the provision of an effective user interface to the hardware, and the management of processes being executed. The scope of these functions becomes even more complex as processors, file systems, etc., are organized in the form of computer networks. In addition to managing a large set of devices, the operating system must also govern communications between these devices and, in a distributed system, make the network of devices appear as a single entity to multiple users. In the context of this paper, the terms device and resource are used interchangeably.

The purpose of this white paper is to examine the options available for implementing functions associated with a network-wide operating system for Space Station. Operating system design involves the implementation of many functions, with a limited number of options for each function. The combination of these options can lead to drastically different operational characteristics. These characteristics include implementation cost, runtime cost, speed, data capacity (in network OS), reliability, ease of use (level of transparency), and ease of modification (expendability). Unfortunately, the optimization of one characteristic generally leads to degradation of other desired characteristics. The trade study should therefore carefully address desired operational characteristics in order to determine the proper options for implementing OS functions.

What is a distributed operating system? A distributed operating system (DOS) is a relatively new concept which extends the concept of an operating system from individual machines to ones connected through a network. Just as operating systems for individual machines have evolved to the point of providing transparency from multiple users (time sharing), transparency of files (a file may be on one of several disks), and interprocess communication, the goal of designers of distributed operating systems is to extend the notion of resource access transparency and interprocess communication to machines in a network. Before doing so however, several constraints have to be met. These include the development of protocols to assure error-free, timely transmission of messages across the network, assuring that interprocess communication between two hosts is possible, and if so, implementing an efficient algorithm for interprocess communication, and finally, implementing the protocol for resource access transparency.

The major topics discussed in this paper are:

- Requirements of the Space Station Data Management System (DMS)

- Functions of a network-wide operating system

- Verification of customer and operator commands

- Determination of whether an operating system function should be local to each processor in the network or centralized

- The implementation of resource access transparency

- The implementation of an interprocess communication mechanism across the network

- The management of tasks and workloads in processors

- The implementation of functions associated with network communication protocols, such as routing, congestion control, and communications and tracking system)

- Commands may be for immediate execution, for delayed execution, or stored

- Commands may be issued directly to the system or as secondary requests from other commands or subsystem functions

- The DMS shall provide links for command authentication as well as provide data and command privacy

- The SS should be capable of autonomous operation between scheduled periods

- Capable of maintenance with minimum interference with other operations

- Provide for monitoring, checkout, and fault detection

The underlying operating system to support the requirements listed above must be capable of at least the following:

- Handle high volumes of data at real-time speeds

- Transparency — Ability to look upon the network as a single machine. Facilitates sharing of resources and the transfer of a process from one machine to another. If a number of onboard processors are dedicated to a unique function, dynamic allocation of processors may be restricted to non-critical functions

- The operating system or an application program must be capable of verifying and scheduling user commands as well as ensure privacy of same

- A means of monitoring network operations must be provided

- The operating system must be extendable and reliable

2.1.3.1 OPERATING SYSTEM FUNCTIONS

The SSOS will need to perform all the functions associated with traditional, single machine operating systems. These functions include task management, memory management, file management, input/output (I/O) management, managing communication between processes, and the user interface. Since the Space Station (SS) will consist of a number of local area networks (LANs), the management of communication between machines is an added issue.

- Task management is the allocation of resources to accomplish a given task. Time sharing of a CPU and the allocation of processors to tasks in a network are examples of task management.

- Memory management is the allocation of required memory to programs. Due to problems of concurrency control and efficiency, memory is generally not shared between processors in a network, making memory management a problem of individual machines in the network.

- File management is the ability to support the creation, opening, closing, reading, and writing of files. The file manager must regulate access to files and prevent multiple users from concurrently modifying a file.

- The operating system must manage communication between programs and the initiation of subtasks. In a network, if a process requires communication with another machine, the SSOS should be capable of executing another process while waiting for information for the first.

- The operating system also provides the user interface to a machine. A user (operator or customer) of the SSOS should be able to access a remote resource with commands of the same complexity as for accessing local resources, i.e., the network must be transparent to the user.

Finally, in a network, a number of unique, communications-oriented problems arise, including addressing, sequencing, routing, buffering, flow control, error control, and security. These issues fall between layers 2 through 6 of the standard 7 layer International Standards Organization (ISO) reference model of open systems interconnection (OSI). Layer 7 tasks are addressed in the implementation of traditional operating system functions. Layer 1 functions, addressing the physical media which comprise the network, are discussed in a separate white paper (1.7.1.1). A description of the OSI model is provided in the final section of the paper for those who may be unfamiliar with the concepts.

In addition to the functions listed above, several other management-related functions are necessary in order to meet the DMS requirements. These may be implemented within the SSOS or as application programs utilized by the SSOS. The functions include command verification, scheduling user access to SS resources, network configuration management, network status monitoring, error detection, and error recovery. Of these, only command verification and scheduling of resources is addressed in this options paper. Discussions of the other functions may be found in references 3-5.

2.1.3.1.1 REFERENCES - OPERATING SYSTEM REQUIREMENTS

1. "Space Station Reference Configuration Description", JSC-19989, NASA Johnson Space Center, Houston, Texas, August 1984

2. "Space Station Definition and Preliminary Design - Request for Proposal," NASA, September 1984

3. Marjory J. Johnson, "Network Control," RIACS TR 84.4, NASA Ames Research Center, July 1984

4. Larry Wittie, "Specifications for Space Station Network Operating System for Data Communication Server," November 1984

5. Raymond Turner, "Network Operating System Functional Requirements," March 1985

6. Edwin C. Foudriat, et. al., "An Operating System For Future Aerospace Vehicle Computer Systems", NASA Tech. Memo 85784, NASA Langley Research Center, Hampton, VA, April 1984

2.1.3.2 SPACE STATION ONBOARD CONFIGURATION

As obtained from the Phase B RFP, Figure 1 shows a possible onboard configuration of Space Station. The onboard configuration will consist of at least two independent networks, "housekeeping" and "payload". As shown in Figure 1, within each individual local area network (LAN), application programs which control the vehicle (core) and payload operations will execute in any of several processors (labeled SDP). The SDPs will be interconnected by a local area network (LAN) that consists of a number of network interface devices (IDs) and a communication medium. Multiple LANs configurations may exist due to the placement of an individual LAN within each of the SS modules, or due to the implementation of separate core and payload network, among other possible combinations. Individual LANs will be interconnected by bridges or gateways. Telecommunication to external systems, ground networks, or orbital free flyers will take place through gateways. For the purposes of this paper, the IDs in Figure 1 will be referred to as Network Interface Units (NIUs) to be consistent with the NIU options paper (Network Interface Devices, 1.7.1.2).

2.1.3.3 SCOPE OF THE ONBOARD DISTRIBUTED OPERATING SYSTEM

The requirements of the onboard component of the SS data management system as well of an onboard network-wide operating system have been introduced. This sections describes the scope and terminology of this white paper before design options for implementing the functions of an onboard network-wide operating system are presented.

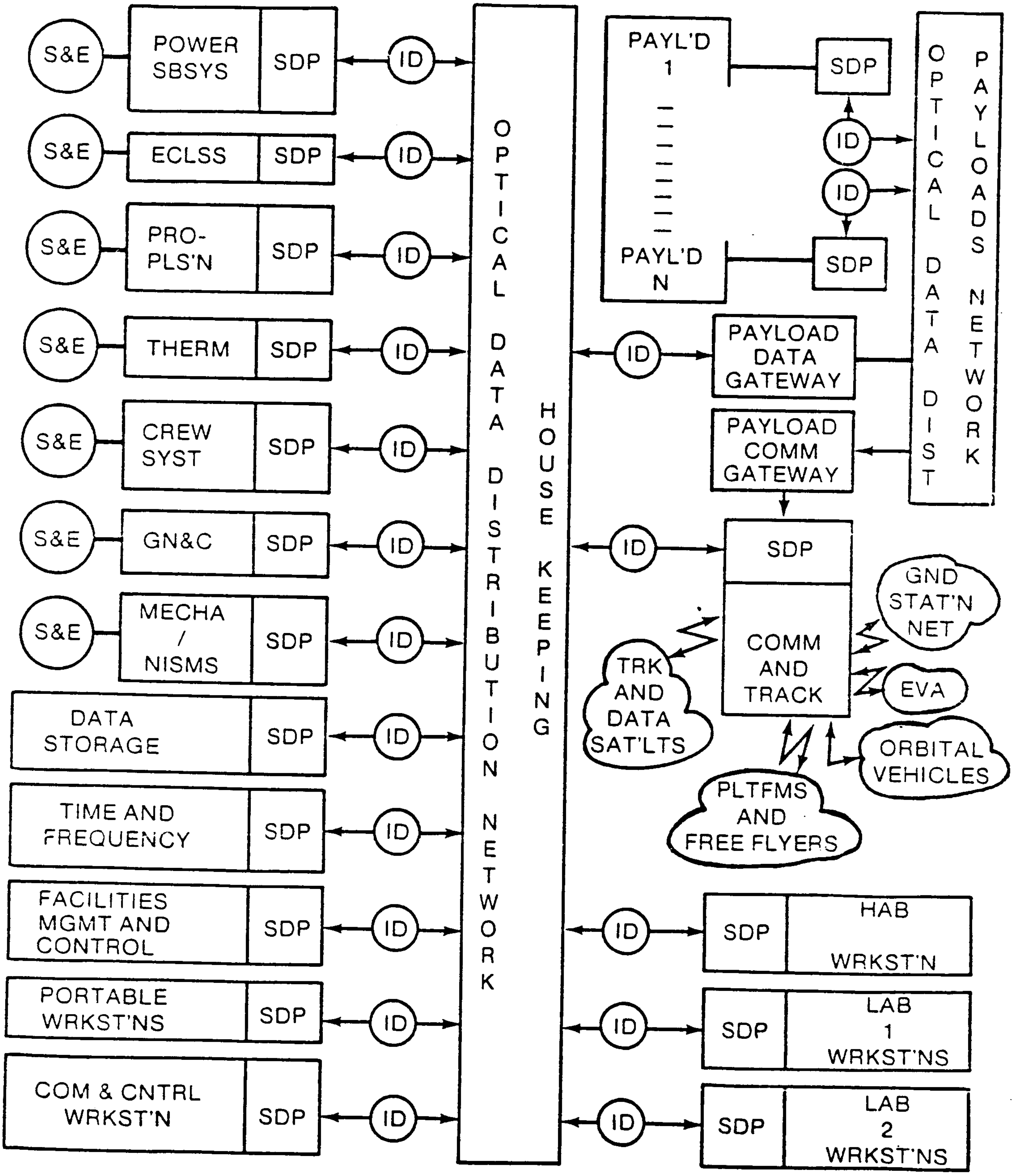


Figure 1. Possible Space Station Onboard Configuration

Source: SS RFP

TERMINOLOGY AND SCOPE

DOS - The Space Station network-wide operating system. Functionally, the DOS allows a user to look upon the onboard network as a single entity. This gives the advantage of resource transparency, both with peripherals and processors. Physically, the DOS will exist partially in the NIUs and partially in the attached SDPs, where the SDPs may be performing core or payload functions. Customers may also provide their own host processors, which are then connected to the network through an onboard NIU. The NIU will perform as many of the network communication functions as feasible, both to offload the SDPs and to provide as much transparency of the network as possible to the customers who wish to develop their own processors. It is important to note that some authors refer to the component of the DOS which executes in the NIU as the Network Operating System (NOS). Such a definition has been utilized in the Network Interface Devices options paper.

A given DOS function may be reside physically in every NIU/SDP (distributed), in particular NIUs/SDPs (centralized), or reside partially at every NIU/SDP as well as at a centralized location (partially distributed). While the terms centralized, distributed, and partially distributed are used frequently in reference to the residency of a given DOS function, no attempt is made to divide a function between NIUs and SDPs. The division of network communication functions between NIUs and SDPs is addressed in the final section of this white paper as well in the Network Interface Devices option paper (1.7.1.2).

In summary, the remainder of this paper describes options for implementing functions required of the onboard DOS.

2.1.3.4 DESIGN ISSUES FOR GLOBAL OPERATING SYSTEMS

The concepts of DOS has been introduced by the previous section. This section will address the implementation of functions of such a DOS. The functions include:

- Verification of customer and operator commands (this function may be thought of as an application function rather than as part of the DOS)

- The determination of whether DOS function is to be centralized, distributed, or partially distributed

- The implementation of resource access transparency

- The management of tasks and workloads in processors

- The mechanism by which the application program accesses network resources, be it a need to access a remote sensor or to set up and maintain an interactive session with a remote process

- The implementation of specific functions associated with network communication protocols, including scheduling, routing, congestion control, flow control, security, and segmentation of messages

- The distribution of network communication functions between NIUs and hosts

2.1.3.4.1 GROUND/SPACE CUSTOMER INTERFACE

Ground customers will communicate with SS onboard systems through the ground/space customer interface. This interface may be implemented through the use of formatted telecommand and telemetry packets. The CCSDS reference model provides an example of such formatted packets. The operator/customer interface functions of concern to the DOS are the questions of "where does command verification take place" and "where does the execution of the command take place." The former is a question of safety while the latter is a question of privacy.

2.1.3.4.1.1 Command Verification

Telecommand verification is a necessity for ensuring that such commands will not cause the payload for which the command is intended to take action as to interfere with other operations or to present a safety hazard. The options for ground sourced telecommand verification are (1) on the ground before the command is transmitted from ground to SS or (2) to do the verification onboard. Ground verification will be cheaper, potentially faster due to more powerful processors, be of better numerical accuracy again due to more powerful processors, but must deal with older or insufficient status information than would be available onboard. A telecommand which is verified on the ground may be transmitted directly to a payload processor if it is found to be unrestricted (i.e, the command does not present any potential safety or interference problems).

Onboard command verification will be necessary for commands generated by onboard operators and customers. Since a verification capability will exist onboard anyway, the trade study must determine whether all commands should be verified onboard versus any ground pre-filtering of ground sourced commands.

2.1.3.4.1.2 Processing of Commands

Once a command is found to be unrestricted, the remaining question is "where is the command executed?" Telecommands may be parsed and executed by a designated SDP (standard data processor) or be delivered directly to the payload processor. The former allows the customer to take advantage of services provided by SS while the latter assures privacy. The customer may have the option of choosing between the two techniques mentioned above.

If a customer interface SDP approach is followed, the commands will have to be carefully determined. For example, such a command may be "at intervals of 'X' seconds, issue command 'Y' to payload 'Z'." The SDP will be responsible for issuing 'Y' to 'Z' at the specified intervals. However, as a side effect, a SDP telecommand processor allows the DOS to be aware of the load in each payload processor.

2.1.3.4.2 SHOULD A TASK BE DISTRIBUTED OR CENTRALIZED?

A DOS must be capable of memory management, file and I/O management, task management and user interface in addition to communication between processes (in a single machine) or between machines (in a network). These functions may be distributed among the processors in the network, or be managed by a dedicated processor. For the purposes of this report, the terms machine and processor are used interchangeably, and may refer to NIUs, SDPs, or both. In addition, the terms "SDP" and "host" are also used interchangeably.

2.1.3.4.2.1 Memory Management and Local Process Control

Memory management is the allocation of memory resources between programs which are running on a given machine. The existing operating systems running on individual hosts can be utilized for this function, making memory management local to each machine. Similarly, communication between processes executing in a given machine is a function handled locally. These functions can be implemented using traditional techniques.

2.1.3.4.2.2 The User Interface and Task Management

The user interface and network task management may be a distributed or centralized function. The user interface for onboard users will consist of a series of packages such as command line interpretation, graphics display software, debugging tools, and monitoring functions among others. Multiple user interface processors will provide ease of access to the system at the cost of replicated software.

Distributed

When distributed, a user may make a request through any of the host processors. The existing user interface capabilities of the existing operating systems in the hosts may again be utilized. However, software for processing user commands must be included in every processor equipped with an user interface capability. The actual processing of the command may involve invoking a process on another machine, as is the case in processing the command "Execute process 'X' at payload 'Y' at time intervals of 'Z'." Thus, task management (allocation of resources) is also handled locally.

Diverging slightly, one of the factors which is of importance in determining whether to centralize the task management function is the extent of capabilities of the DOS. Specifically, should a processor be capable of requesting another processor to handle one of its tasks? This capability will be useful, for example, in temporarily transferring the processes executing on a machine scheduled for maintenance. Notice that this scheme implies that each of the host processors must know the location and workloads of other processors in order to determine the appropriate processor to which processes are to be transferred. This may involve high overhead due to the need to update messages stating "I am processor X and this is my workload".

Since only non-time critical processes are likely to be distributed, an alternate approach is for a local host to send a broadcast message or selectively poll remote hosts and request load information when it wishes to distribute its workload. Upon receiving replies, the local host can determine the appropriate remote host for executing the process. This scheme will require update messages only when absolutely necessary and yet ensure that user interface and task management functions are local and distributed.

The democratic scheme outlined above does not consider the issue of priority. If an unexpected, but time critical situation should arise, such as a failure of some kind, a processor must be assigned to a task in a dynamic manner. Each host must therefore have a criterion based on the priority of a broadcast request for considering itself to be busy when polled. The other alternative is to dedicate a machine for handling unexpected situations. Doing so, however, would require a careful study to determine the maximum resource requirements (e.g., processor, memory) of a given abnormal situation.

Centralized

The second option is to leave the user interface and task management function to a dedicated processor. Centralization avoids duplicated software, but raises the question of "What if the dedicated processor becomes unavailable?". The latter problem may be solved by introducing redundancy into the design. From the viewpoint of task management, a centralized user interface makes it possible to maintain some knowledge of the workloads of network processors, enhancing the ability to choose a processor to execute a time critical request. The polling method suggested above may be utilized when it is desired to transfer the processes executing in a machine scheduled for maintenance.

Trade: User Interface and Task Management: Global or Local?

Local (distributed):

o Robust and reliable

o Useful if need for processing on remote hosts is limited

o Will require replication of software in each processor involved

o Loads in other processors are not known

Centralized:

o Useful if a great workload is being imposed on a limited set of resources (allows balance of workload among the processors)

- Can promptly assign a remote host to an emergency request as all workloads are known

- State of manager is unreliable, so redundancy is required

- Task management can coexist with the user interface if processors are not permitted to transfer processes to other machines

A compromise between a fully distributed user interface and a centralized one is a partially distributed one. Here, processors capable of functioning as the user interface will be dispersed throughout SS to provide ease of access to the onboard users. However, such a function will not be totally distributed, thus saving storage space.

2.1.3.4.2.3 File Management

The file management function can be divided between a centralized processor and the hosts of the network. Assuming that files common to a network can be accessed by any member of that network, a dedicated processor may maintain file consistency. This processor may be the file server itself if the server is capable of such a function. The individual processors provide the environment for editing.

2.1.3.4.2.4 I/O Management

Local I/O management will be handled by the individual processors. The management of shared output resources, however, is appropriate for centralization. The term "shared output resources" refers to stand-alone devices such as teletypes and printers. Since the onboard utilization of such devices will be very limited, a simple queue management system at the NIU interfacing the output device to the network may be sufficient.

Summary of the functions as local or centralized:

Local — Memory management, local process management, file management, local I/O management, local task management

Optional — User interface, network task management

Centralized — Management of shared I/O resources and files

Optional — User interface, network task management

Partially — Optional — Network task management Distributed

2.1.3.4.2.5 Address and Routing Tables

In addition to DOS functions, the trade study must address the issue of where necessary information is to be stored. In particular, the DOS will have a need to access various address tables and routing tables during its operation. Address tables will be required for sensors, effectors, and active processes, as well as for frequently accessed variables and other data. These tables are required to facilitate interprocess communication. For example, any monitoring function provided for SS users will need to contact the system being monitored in order to obtain necessary sensor values and other data. However, address tables will be required in order to determine the location of the desired process and data. The contents of these tables are discussed in later sections.

The given table may be centralized, partially distributed, or fully distributed.

- Centralized tables save memory but introduces a finite access time to determine an address or route, in addition to the need for redundancy.

- To reduce the need to access the central source to determine an address, a local cache may be utilized. A central source may maintain a table of all addresses, but the addresses of frequently accessed resources will be stored locally.

- A fully distributed approach trades local memory space for potentially faster access to addresses and routes.

2.1.3.4.3 RESOURCE ACCESS TRANSPARENCY

Files, output devices, sensors, and effectors are examples of resources often shared between the processors in a network. Further, multiple copies of such resources may be physically dispersed throughout the network. In a non-transparent system, it is necessary for the user to explicitly address a resource. A file access may be "get this file from that file server." In a distributed system, the user only needs to specify "get this file." The DOS determines the location of the file and accesses it.

Resource transparency for static resources can be accomplished by storing tables mapping logical names to physical locations. On the non-technical side, there is a need to provide the user with logical names for a class of resource and physical locations if appropriate. If the user has the need to access a given class of device, but is not aware names used within the network, a list of devices and associated logical names will be very helpful. This section will present options for implementing address transparency (of processes in particular) and for implementing transparency of dynamically changing resources, with the specific example of file access.

Along with knowing the location of a resource, it is also necessary to know whether a resource is active or down. This information may be incorporated as a flag beside each name in an address table. When such a resource as a processor becomes active, it broadcasts a LOGON message stating its location and function. Similarly, a LOGOFF message is issued when a resource goes down. Any sudden crashes will be detected in due time and will result in the issuing of a LOGOFF message by a second party.

PROCESS	NET	NIU ADDR	NIU LINE
ECLSS	–	1	1
GN&C	–	2	1
Payload 1	2	1	1
Payload 2	2	1	2
Ground Gateway	–	3	1

Figure 2a. Address Table Covering All Onboard Hosts

PROCESS	NET	NIU	LINE
ECLSS		1	1
GN&C		2	1
Payload Gateway		3	1
Ground Gateway		4	1

Table for Core Network

PROCESS	NET	NIU	LINE
Payload 1		1	1
Payload 2		1	2
Core Gateway		2	1
Ground Gateway	1	4	1

Table for Payload Network

Figure 2b. Unique Address Table Per Network

2.1.3.4.3.1 Process Address Transparency

In addition to achieving transparency of physical resources such as sensors, effectors, and files, the processes running within the network must also be transparent. A application process must be capable of contacting any other process by logical name rather than having to specify the physical location and path. Process address transparency can be implemented through address tables in SDPs, NIUs, or both. The trade study should weigh the advantages and disadvantages of the following addressing schemes:

- Each NIU contains the address of every process in SS, including processes which may be active in other networks (Figure 2a). This has the advantage of a potential increase in access speeds and certainty regarding the location of any process, but has the disadvantage of large overhead and makes the addition of a new processes within the network more complex (i.e., a broadcast message must be sent to every NIU on every onboard network stating the location of the new process).

- An alternative technique would take advantage of the fact that the number of onboard networks will be limited. In this scheme, a host/NIU in a given network contains physical addresses for only the processes local to that network (Figure 2b). When an application requests that a message be sent to a process, the table of local processes is searched for the requested process. If the process is not found, the request is sent to the gateway(s)/bridge(s). The portion of the gateway which belongs to the next network contains address tables for that network, which is then checked. In this manner, the message is sent to all the networks of the SS until the process is found. This scheme, while efficient in its use of storage resources, can waste bandwidth and time by its need to search all onboard networks. If the SS is to contain only a small number of networks, and if the need for communication between hosts in separate networks is limited, the second scheme may be appropriate. Note that this scheme has the property of poor response to bad news. If a user-requested process does not exist, that news will be slow in reaching the sender. In addition, this scheme reduces the size of the message when a transmission is intended for another network. This is due to the need for the logical process name to be carried with the message in order to search the address table of the next network. As an supplementary solution, a cache of important internetwork addresses may be maintained within each NIU.

o A third approach is again to have addresses for only local processes, but rather than automatically sending the request to the gateway, a message is broadcast asking for the location of the desired process. If the requested process exists, the bridge possessing knowledge of the requested process sends an acknowledgement to the requesting application. This scheme provides better feedback to the sender, but is wasteful of bandwidth and slow as well.

o A fourth approach is to have addresses for only local processes, but also to know of a central source whether other desired addresses may be found. A cache of important internetwork addresses may also be maintained. Methods two and three above may be involve wasted time and bandwidth, but do not require a central storage facility for address tables. Regardless, method 4 may be preferred over 2 or 3 due to the potential difference in speed.

2.1.3.4.3.2 File Access Transparency

The term file access transparency is misleading. In a distributed system, not only should the DOS be capable of accessing a file given only its logical name (logical path), but the DOS should also be capable of several other functions, including data access control, concurrency control, serializing transactions, deadlock control, and recovery to a consistent state in the event of failures (1). A number of different techniques are employed by existing distributed file systems for implementing the above-mentioned properties. These existing systems include XDFS (XEROX Distributed File System), LOCUS, MULTICS, DEMOS, and CFS (Cambridge File Server) among several others (1). The options for implementing the mentioned properties will first be described before the LOCUS method (2) for implementing access transparency (logical names to physical locations) is presented.

DATA ACCESS CONTROL

The unit of data access can be through a file, sequential subset of a file, a page (used by LOCUS), or a subset of page. Access control is necessary to prevent unauthorized access. The two techniques which have been employed to control access are capability based and identity based systems. In a capability based approach, a user must have an explicit permission to access an object (in this case, a file), while in a identity based scheme, the user's identity is also checked in addition to capabilities in order to provide better security.

CONCURRENCY CONTROL AND SERIALIZATION

Concurrency control allows transaction-type access to shared files. This function is.generally a part of the file system of the operating system, but can be left to the user (XDFS)(1). Concurrency control is the process of locking access to files or pages (i.e., the unit of storage allocation) in order to preserve consistency. Control is generally implemented to follow the following two guidelines:

1. A file/page may be read if it is not being written;

2. A file/page may be written if it is not being read or written.

SERIALIZING is the process of ensuring control over several simultaneous requests for file access. The two techniques, which are described in detail in section 2.1.1, are two-phase locking and timestamp ordering. In the former, a user transaction must acquire locks on the entity to be accessed. Once all required locks have been obtained, the transaction is carried out, after which the locks are released for use by other transactions. The two-phase lock approach maintains consistency by ensuring that a file or database is not updated until the transaction is fully authorized to do so. It is possible for the two-phase locking method to be unfair (3). The timestamp ordering approach is a fair technique which provides access to resources on a FIFO (first-in, first-out) basis. The server machine stamps the arrival time of each request. The timestamp approach is also free of deadlocks, which may occur with the two-phase locking approach.

DEADLOCKS

A deadlock may occur if two transactions require access to each of two files and each transaction has a lock on one of the files. The file system may handle deadlocks in the following ways:

1. Deadlocks may be prevented if all required files are declared before a transaction begins.

2. Deadlocks may be detected and resolved by aborting one transaction in favor of another.

3. While the system is not congested, a timeout mechanism for waiting on a lock may be employed. If a transaction is not able to obtain the required locks within a declared period, that transaction aborts and tries again at a later time.

4. Timestamps may be combined with two-phase locking to allow older transactions more time to acquire the necessary locks than younger transactions.

5. A priority based scheme may be employed, with one of the other techniques serving as a backup to resolve conflicts between transactions with equal priorities.

6. Deadlock handling may be left to the users.

STRUCTURE OF FILE SYSTEMS

File systems may be organized as a multilevel tree of fixed-size blocks, of variable-size blocks, or as a set of fixed-size blocks. To implement RECOVERABILITY, the most common solution is the SHADOW-PAGE TECHNIQUE. In this scheme, only file pages which have been modified are written into free blocks on the storage device. The file page map is then updated to indicate the new mapping of files to storage blocks.

Requests for file access may follow one of the following protocols:

1. send request/receive response
2. send request/receive response/send acknowledgement
3. a single step, no-wait send

In addition, a file access protocol may be specialized (2) or be the same as the general purpose interprocess communication (IPC) facility used for communication between processes on different machines (4) These techniques are discussed in detail in the following section.

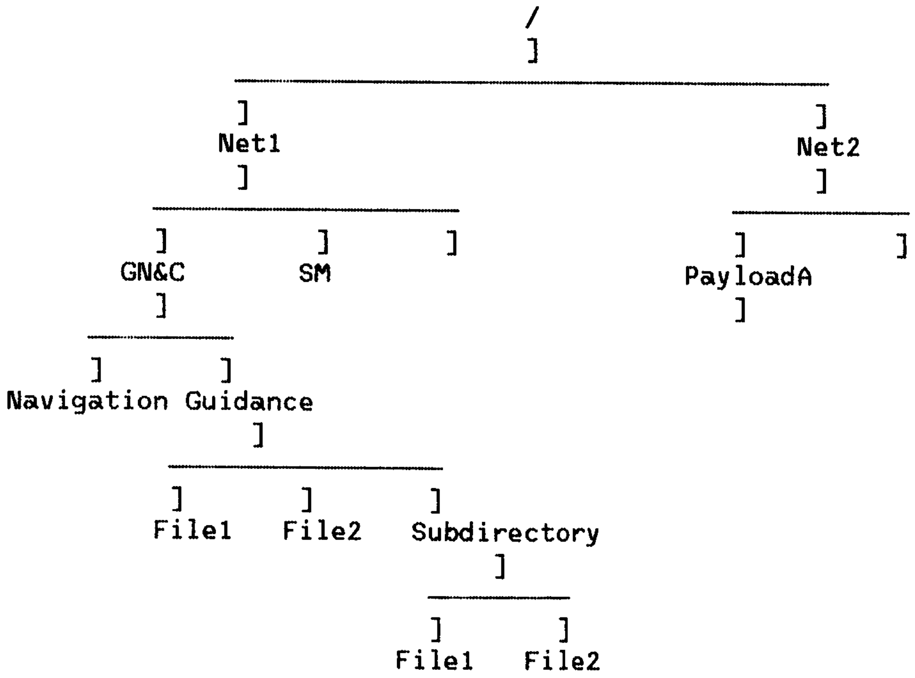


Figure 3. Hierarchical File System

THE FILE ORGANIZATION SCHEME OF LOCUS

In this subsection, the file organization scheme of the LOCUS distributed operating system is presented as a technique to enable the user to specify a file by logical name without specifying the physical location. LOCUS is a functioning DOS developed at UCLA and is based on the UNIX operating system. LOCUS is approximately one-third larger than UNIX.

LOCUS uses an UNIX-like tree structured directory system for its files (Figure 3). The location of the root of the tree (/) is known to every file storage site. Each directory represents a collection of files which form a filegroup. A filegroup may contain subdirectories. LOCUS treats each such filegroup as a separate entity. Each logical filegroup has several physical locations allocated to it. A given file in that subgroup may be present in any of the designated locations, including multiple locations. Each file is uniquely represented by (logical file group number, file descriptor number). Current file operations include open, create, read, write, commit, close, and unlink. LOCUS maintains file consistency across multiple copies and uses a shadow-paging commit protocol for writes. Multiple copies are supported for easier and faster access, with the penalty of a complicated consistency control mechanism.

As stated before, the user specifies a desired file through its pathname. The use of pathnames and directories has special appeal for the Space Station. Each major function, such as GN&C can have its own directory, with all pertinent files stored within that directory. Note that individual files may be spread across different physical locations. A user need only know the name of the desired file and its category. Such helpful UNIX facilities as dtree (draws the subtree of directories and files starting from the current working directory), and whereis (provides the pathname to a user-specified file) can be provided to aid the user.

LOCUS requires that only the location of the root directory and the location of current working directory be known to every site. Once the user requests a file, the pathname is expanded as described below.

If the pathname begins at the root, the root directory is searched for the next pathname component. Note that pathnames consist of directory names, with the last entry being the name of the desired file. If the root directory exists at a physical location different from the current one, the directory may either be paged across the network and searched at the local site or be searched at the remote site. Once the location of the next directory is determined, it is accessed and searched for the next element in the pathname. This continues until the desired file is located. If the directory tree is deeply nested, accessing a file may be a slow process. To obtain better performance, the following modification of the LOCUS approach may be appropriate.

Rather than storing only the location of the root and current working directories at each site, the locations of a number of directories may be stored in tables at each node in the network. When a file access request is received, the pathname is searched, component by component, within the local table. At the point when a directory is not listed in the table, the LOCUS method of pathname expansion can be utilized.

Consider the following example. Assume that the local tables contains the locations of the high level directories root, designated by "/" Net1, and GN&C. If a user issues an access request for the file specified by /Net1/GN&C/Guidance/NeededFile, the search of the local table expands the pathname until Guidance is encountered. At this point, the GN&C directory is accessed by the LOCUS method and the location of Guidance is determined. A search of the Guidance directory reveals the location of NeededFile.

Another option is presented in (5). Here, to differentiate between files used regularly by time critical functions and others, a two part file system is proposed - a real time component and a time shared component. The user must declare all system requirements (file contiguity, communications bandwidth, etc.) during the creation of and while opening real-time files. A real-time process is required to open all necessary files before starting execution. The time-shared component follows a UNIX like approach. It is suggested that an exception processing capability be added to handle communication or node failures.

2.1.3.4.3.3 References

1. Anna Hac, "Distributed File Systems - A Survey", Operating Systems Review, Vol, No. 1, pp. 15-17, January 1985

2. Bruce Walker, Et. Al., "The LOCUS Distributed Operating System", Proc. Ninth ACM Symposium on Operating Systems Principles, pp 49-69, October, 1983

3. Philip Bernstein and Nathan Goodman, "Concurrency Control in Distributed Database Systems", Computing Surveys, Vol. 13, No. 2, June 1981

4. David Cheriton and Willy Zwaenepoel, "The Distributed V Kernel and its Performance for Diskless Workstations," Proc. Ninth ACM Symposium on Operating System Principles, pp. 129-139, October, 1983.

5. Edwin C. Foudriat, et. al., "An Operating System For Future Aerospace Vehicle Computer Systems", NASA Tech. Memo 85784, NASA Langley Research Center, Hampton, VA, April 1984

2.1.3.4.4 TASK MANAGEMENT - SHOULD THERE BE DYNAMIC TRANSFER OF PROCESSES?

New tasks, ones which have not been assigned a processor, may occur in one of three ways:

- A new memory load

- New user/customer applications which require a processor

- A processor requests that a process be transferred to another machine

The last of these cases was introduced in section 2.1.3.4.2.2. The ability to handle the first two cases is necessary, but not the third. This section will examine this third case in detail and present the trades of having such a facility.

Possible scenarios at the time of a request to transfer a process to a remote processor:

- The ideal case is a process for which all pertinent data and execution software is available at the remote processor. The only costs would be initial transfer of the process to the remote machine and the cost of transporting the results back to the local machine. The ideal case applies, for example, to redundant machines.

- The next case is the existence of execution software at the remote machine with pertinent data at the local machine. If the data can be copied to the remote machine, there is increased delay due to the transfer of data and an increase in the needs for storage at the remote site. If the amount of data is large, the process may execute at the remote machine, but accesses data from the local machine through the network. This would involve high delays due to network transmissions. It is important to recognize that certain data may be very difficult to transfer, if at all. Consider, for example, the prospect of transferring a process which requires data from sensors physically connected to the local machine.

- The third case occurs when both execution software as well as data need to be transferred. Assuming that the remote machine has the necessary CPU and memory resources, this "total" transfer may not present major problems for Space Station, the reason being the projected use of common language(s) and processor(s). Of course, the obvious overhead is the cost of network transmission of a large file of execution software.

In addition to those mentioned above, another problem encountered in transferring processes is not in the transfer itself, but in updating address tables. Before this update is completed, a second process may attempt to initiate a conversation with the process which was moved. A mechanism for re-routing such requests is therefore necessary. Such a mechanism must also re-route any conversations which may have been in effect before the transfer.

The next question is the determination of when a process should be transferred to a remote machine.

The local machine may temporarily be overloaded, a process may arrive which has not been assigned to a machine, a machine may be due for maintenance, but is not part of a redundant set, and so on. In addition to the "emergency" situations mentioned above, a process may be relocated simply to make better use of available resources. The DOS must include an algorithm for assessing the benefits versus costs of moving a process. Such an algorithm would need to take into the account the loads (CPU and memory utilization) of the local and remote machine, the needs of the process at hand, and the network conditions among other possible factors.

Certainly, the ability to transfer a process is not without overhead. The trade study should carefully examine the needs for such a dynamic transferring capability. The alternative to dynamic transfers is to design the Onboard system is such a way that situations requiring a dynamic transfer capability do not arise. Such techniques include assigning every possible task to a machine, having sufficient resource redundancy, and ensuring that any given machine will not be overloaded. Clearly, such techniques will be wasteful of resources since they must be designed with worst-case situations in mind. The trades of having a dynamic transfer capability are examined below.

2.1.3.4.4.1 Dynamic Transfer of Processes Vs. Confined Processes

The advantages and disadvantages of confining processes to the machine where they exist as opposed to the ability to transfer a process to a remote machine are discussed below:

Advantages of confined processes:

- Reduces complexity of OS, reduces network overhead
- Predictable execution times
- Better security by reducing possibility of user interactions

The disadvantages of confined processes include:

- Increased hardware costs, power costs, space requirements due to wasted resources in designing to meet worst-case conditions
- Potential difficulty of reconfiguration and need for redundancy

The advantages of dynamic transfers include:

- Freedom and extendability
- Better use of resources, requires a smaller number of resources
- Allows graceful degradation of the network, easier to reconfigure

The disadvantages of dynamic transfers include:

- Difficult to test and verify
- Initial design of DOS more complex
- Increased checking by the DOS — For example, should not transfer a process into a machine being utilized by a customer requiring a high level of security
- Immature technology
- High cost of development

2.1.3.4.5 INTERPROCESS COMMUNICATION MECHANISMS

There will be much need for communication between processors in the SS network. Conceptually, such communication may be in one of three forms:

- o Accessing data from an non-processing device, such as a sensor
- o Submitting commands, such that by the user interface processor to the payload or submitting a job to an output device
- o Calling on a process on a remote machine to compute and supply a result which is then utilized by the caller machine to finish its task

As with the discussion of task management, the first two cases are necessary, with the third being suitable as a growth item since such a capability may be useful in a payload environment.

The options for implementing the ability to talk to a remote machine are:

1. A Space Shuttle-like approach where the code is written to transport a necessary sensor value to the process which requires it at the instant when it is required. This approach will make the initial design much more difficult and will also be difficult to modify. For these reasons this method will not be considered further.

2. The second technique is to have a set of commands for interprocess communication. Such a command may be GetSensorValue(Sensor). The advantages of this technique are that the application programmer knows the syntax of the commands and that tables of only resources which may be accessed are necessary.

3. The third technique is to have a general IPC facility. With this approach an application process on one machine directly contacts a process on the remote machine. The DOS only serves as a facility to transport a message from the local machine to the remote one. While the second technique also requires an IPC mechanism and addresses the same issues as the third, the third approach provides the ability for any process to talk to any other. However, overhead will increase with the need to maintain the location of all processes. In addition, during development of application processes, there is a need to maintain contact with other developers to ensure type matching, proper inputs and outputs, etc.

While the implementation details of an IPC facility are beyond the scope of this report, it may be beneficial to point out that there are several methods by which an IPC facility may be implemented. Two frequently encountered approaches are message passing (2) and remote procedure calls (3,4).

The issues addressed by these techniques include:

- The facility for creating a process on a remote machine

- The facility for properly initializing the remote process

- The interprocess communication mechanism (IPC). The IPC across the network should have the same semantics as the IPC for a single machine.

- The ability to maintain a consistent state in the event of failures in portions of the network.

- The ability to reflect error conditions across network boundaries.

The IPC mechanisms employed for communication across a network are extensions of facilities for IPC within a single machine. The conceptual view of a remote procedure call is shown in Figure 4. The interested reader may refer to (3) for further details.

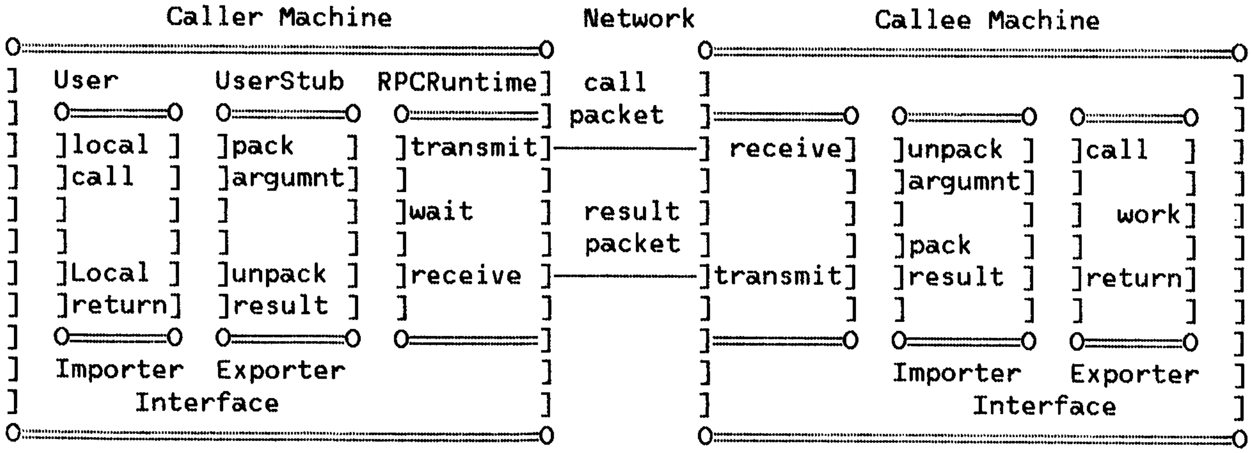


Figure 4: Remote Procedure Call
Source: Birrell and Nelson

2.1.3.4.5.1 References: Network Interprocess Communication Techniques

1. John A. Stankovic, "A Perspective on Distributed Computer Systems", IEEE Transactions on Computers, Vol. C-33, No. 12, December 1984

2. David Cheriton and Willy Zwaenepoel, "The Distributed V Kernel and its Performance for Diskless Workstations," Proc. of the ninth ACM Symposium on Operating System Principles, October 1983, pp. 129-139

3. Andrew Birrell and Bruce Nelson, "Implementing Remote Procedure Calls", ACM Transactions on Computer Systems, Vol. 2, No. 1, Feb. 1984, pp. 39-59

4. Bruce Walker, Et. Al., "The LOCUS Distributed Operating System", Proc. Ninth ACM Symposium on Operating Systems Principles, Dec. 1983, pp. 49-69.

5. Barbara Liskov, "On Linguistic Support for Distributed Programs", IEEE Transactions on Software Engineering, Vol. SE-8, No. 3, May 1982, pp. 203-210

2.1.3.4.5.2 Moving Data to the Process

The primary need for interprocess communication is for accessing data, whether in the form of computed results or raw sensor readings. If such data is accessed on a regular basis, much bandwidth can be wasted due to the need for every process requiring that data to contact the process governing the data. An easier, more efficient approach is to make the data available through a database or through broadcasting the information. This may be especially true for ancillary data.

If a broadcast approach is followed, all data which is required by SS processes on a frequent basis is broadcast regularly by the process which owns the data. While the broadcast itself will consume quite a bit of bandwidth, processes needing the data need not send explicit messages requesting the data. This may not only be faster, but also reduces the size of address tables within each NIU. Note that the size of any centralized address table will not be reduced for reliability reasons. Further, an IPC facility will be necessary as a backup since broadcasting is not 100 percent reliable.

The alternative to broadcasting is to send all frequently accessed data to a common database. This again reduces the size of address tables within each NIU and also allows a process to request and obtain all the data values it needs with a single request. This technique may also require an IPC mechanism as a backup since a centralized database may itself be unreliable. Further, traffic in the vicinity of the database will be heavy, both due to updated data values and requests for the data values.

2.1.3.4.6 NETWORK PROTOCOLS - FUNCTIONS

This section provides a summary of the issues which arise in implementing the network-oriented communication functions of a DOS. These issues include scheduling, security, segmentation, routing, congestion control, and flow control. Another issue, addressing, has been discussed in section 2.1.3.4.3.1. Further, the type(s) of service (connection-oriented vs. connectionless) to be provided by the network protocol is discussed in the Network Interface Devices (1.7.1.2) white paper.

2.1.3.4.6.1 Scheduling

Scheduling, or prioritizing, is a necessity in a network with limited resources. Scheduling will be required for handling a number of situations, including time dependent requests and requests dependent upon the order of execution. In a priority-based system, there is a danger of deadlock if requests in contention have equal priorities. A priority scheme might be based on the type of process (read, write, etc.), the user, or importance of the data to the requesting process. The algorithm of timestamp ordering, developed primarily for concurrency control, may provide a means for ordering user requests. Further, the operating system will need to maintain concurrency control over file systems and common databases. Concurrency control algorithms, including timestamp ordering and two phase locking are discussed in section 2.1.1 (Distributed Database Management).

A scheduling algorithm may be based on several characteristics:

- Type and amount of state information used:
  - Queue lengths
  - CPU utilization
  - Amount of free memory
  - Estimated average response time

- How and when update information is transmitted

- Degree and type of cooperation between distributed scheduling entities (i.e., are decisions made with local or global information?)

- How often is the scheduling algorithm invoked?

- Adaptability of the algorithm

- Stability of the algorithm

Comments on characteristics mentioned above:

- State Information: Is the data too old? Is it accurate?

- Invoking the scheduler: Too often, overhead, too infrequent, cannot adapt quickly to changing conditions

- o adaptive scheduler:
  - light loads — turns off the scheduler, expect a network load monitor
  - medium loads — make full use of scheduler
  - heavy loads — use of scheduler may make the system even slower. Use of bidding (poll processors for an available one) or reverse bidding (a free processor requests work) is a second option

- o Measuring network conditions: overhead versus benefit

- o Moving a process to a remote machine: overhead versus benefit

REFERENCE: SCHEDULING ALGORITHM CHARACTERISTICS

1. John A. Stankovic, "A Perspective on Distributed Computer Systems", IEEE Transactions on Computers, Vol. C-33, No. 12, December 1984

2.1.3.4.6.2 Security

The operating system is responsible for providing a protected environment for the execution of operations which require security. This can be accomplished by scheduling secure tasks to specific machines and ensuring that these functions are not subject to dynamic allocation. In addition, the operating system may provide the user with encryption services. The significant portion of the responsibility of ensuring that data is properly encrypted and secured will lie with the user.

2.1.3.4.6.3 Message Routing

Assuming that a mesh topology is not chosen for the SS LAN, the need for routing arises only in internetwork communication. Consider that the SS network consists of five networks connected in a line. It is not sufficient for a host in network 1 to obtain the address of network 5 in order to communicate with a host in 5. There is a need to determine the way to get from network 1 to network 5, hence the need for routing.

The DOS is responsible for routing messages throughout the network to isolate the application programs and hosts from needing to know the topology which interconnects the various parts of the network. The algorithm for determining a route will have tradeoffs with correctness, simplicity, robustness, stability, fairness, and optimality. A variety of routing schemes exist and fall under one of the major categories of static and dynamic routing.

STATIC ROUTING

Static routing algorithms require that a table of network hosts and paths to them be loaded into each node of the network and not changed thereafter. These tables provide information regarding the best path, second best path, third best, and so forth along with a weighting scheme for each path. The node determines the outgoing path based on the generation of a random number and its relation to the weights of paths (e.g., if the best path has a weight of .7 and second .3, a random number between 0 and 7 in a range of 0–10 will result in the choice of the best path). Static routing schemes have the following properties:

- Simple to implement
- Can make good use of alternate routes
- Can give good performance
- Useful only if topology and traffic conditions are stable over time
- Cannot adopt to sudden adverse conditions (congestion, NIU down)

DYNAMIC ROUTING ALGORITHMS

Static routing algorithms have the serious problem of being nonadaptable. Dynamic routing schemes circumvent this problem by basing a route on current network conditions. Dynamic routing schemes may employ a central router or a local router.

In a central routing scheme, a single host in the network functions as the router. All other nodes periodically send messages to the router stating their workloads and the number of packets sent out on each local line by the node. The central router calculates the optimal route from each node in the network to every other node and sends updated routing tables to every node in the network. The central router scheme has the following advantages and disadvantages:

- Generates optimal routes
- Relieves individual NIUs of the task of calculating routes
- Increases network traffic in the form of update packets
- Central scheme always less reliable than a distributed scheme since the central controller may fail
- If a NIU in the network suddenly goes down, the central router scheme is not robust
- Heavy concentration of traffic around the central router
- NIUs close to the central router will get updated routing tables before further NIUs. This may lead to inconsistencies

LOCAL ROUTERS

Local routing algorithms may be based on local traffic conditions (isolated adapting algorithms) or be distributed and based on network-wide traffic conditions. The latter trades increased network traffic (update messages from one NIU to another) for more optimal routes. Tanenbaum (1) discusses the options and trades for each of these two algorithms in detail. Here, only the trades between isolated and distributed algorithms are given.

Isolated Adaptive Algorithm Characteristics:

- Path chosen may only be locally optimal, the next node may have a very long queue or even be down
- Easier to implement and does not result in network traffic overhead

Distributed Routing Algorithms:

- More complex
- Require network traffic overhead for update packets
- Can determine a route in a more optimal manner

Recommendations:

Since the SS configuration and network traffic flow conditions will not change significantly on a regular basis, a static routing table may be sufficient. If a dynamic routing algorithm is desired, the following recommendations are offered:

Since one of the primary requirements of the Space Station data management system is to handle large quantities of data in realtime, isolated algorithms are not recommended. The determination of central router versus a distributed algorithm is more difficult, but a distributed algorithm is recommended for its reliability over a central one and for the characteristic that network traffic overhead due to updates is distributed over the network rather than concentrated at one node, as is the case with the central router scheme.

BROADCASTING

Broadcasting is a useful technique when a message has to reach all the nodes in a network, such as routing information in a central router scheme. The obvious approach of sending such a packet out on every output line other than the one the packet arrived on can be wasteful of bandwidth since the scheme can result in multiple copies of a packet being delivered to a given node. Special care must also be taken to ensure a loop does not result. In addition, the transmission may not reach every node due to such adverse conditions as static during transmission and local congestion in parts of the network. Several algorithms have been developed for efficiently implementing the broadcasting function, and are described in (1).

REFERENCE: ROUTING TECHNIQUES

1. Andrew S. Tanenbaum, Computer Networks, Prentice-Hall, Englewood Cliffs, New Jersey, 1981

2.1.3.4.6.4 Congestion Control

Congestion is the degradation in network performance due to an overload of packets in the network. In a shared-bus LAN, restricted access to the bus is sufficient as a congestion control mechanism in most cases. However, congestion can occur when a particular NIU or gateway is swamped with packets from several different sources. A congestion control scheme will be required to restore order to the network. A number of congestion control algorithms have been developed. These include:

1. Preallocation of resources to avoid congestion

2. Allowing NIUs to discard packets at will sets a timer. The receiver is required to send an acknowledgement before timeout or the message is retransmitted. An acknowledgement will not be received if either the original message never reached its destination or if the acknowledgement is lost during transmission. The sender must maintain a copy of the message in its buffers until it is acknowledged.

The packet discard technique takes advantage of this data link protocol by assuming that a packet will be retransmitted. This technique is sufficient if congestion is not a regular problem, the data being sent is not time critical, and if the congestion is only temporary.

RESTRICTED NETWORK ENTRY

Congestion can be avoided by restricting the number of packets in the network. This can be implemented by the use of permits for entry into the network and controlling the number of permits. This technique has a number of drawbacks which point to its avoidance for Space Station. A NIU can still be swamped, the distribution of permits may not be fair, and a permit may be destroyed, thus reducing the carrying capacity of the network.

CONGESTION CONTROL THROUGH FLOW CONTROL

Flow control algorithms may be used in some cases for congestion control. Tanenbuam (1) argues that optimal flow control cannot be designed when congestion control is a function of that algorithm. However, flow control has been effectively used for congestion control in such networks as the ARPANET and the Canadian public data network, Datapac. Connections between flow control and congestion control are discussed in the next section.

CHOKE PACKETS

All four of the schemes which have been described have the potential for wasting bandwidth. The use of choke packets is a more dynamic technique of congestion control. In this scheme, each NIU maintains of record of the traffic on connected lines. If the utilization of a line is reaching its critical point, every incoming packet is checked to see if it needs to be output on the highly utilized line. If so, a message is sent to the sending host requesting that transmission be reduced by a percentage. If the host does not receive any other choke packets for some time, it reverts to its normal mode of operation. For handling cases of extreme congestion, this scheme has been combined with the packet discard approach. The packet discard backup will be an effective means of fault tolerance in the event that a choke packet does not reach the host for whom it is intended.

1. Channel Queue Limit Scheme: allocate buffers for each output channel. Packets are distinguished by the output queue they must be placed into.

2. Allocate buffers based on hop counts - the number of NIUs traversed by a packet.

Another design issue is the degree of sharing of buffers:

1. fixed, uniform buffering among buffer classes

2. buffer proportional to traffic in each class (no sharing)

3. oversell buffers — sum of buffer limits of each class is larger than the pool

4. Dynamically adjust buffer limits based on traffic flow

Channel Queue Limit (CQL) Scheme:

- Eliminates Performance Degradation at Peak Loads
- Eliminates direct store-and-forward deadlocks (1)
- Optimal sharing best scheme for throughput reasons
- Does not prevent indirect store-and-forward deadlocks

According to Gerla and Klienrock (1), "Some form of CQL flow control is found in every network implementation." The ARPANET protocol contains the following CQL strategy:

Maximum number of buffers (per NIU): 40 Minimum of 2 input buffers and 1 output buffer per line. 10 buffers for reassembly before going to hosts. Maximum buffer limits: Reassembly 20, Output 8, Total Store and Forward 20

— Generally a window scheme that provides for the storage of N messages. A collective acknowledgement is sent once all N messages have arrived. A sender may not place more than N messages into the network until an acknowledgement is received. The sender may eventually time out and retransmit.

— Implementation issues:

— Window sizes — Is error control and loss control to be provided? — Retransmission timeout interval

NETWORK ACCESS FLOW CONTROL

— Throttle external inputs based on internal network conditions.

— Options: Permits for access, Choke packets, and Input buffer limits

Input Buffer Limits

— Limit entry traffic in favor of transit traffic at every NIU. Drop input packets when buffers are full

— Achieves better throughput, but can be unfair

— Buffer allocation may be based on input message throughput versus total message throughput at every node

TRANSPORT LEVEL FLOW CONTROL

— Uses credit window scheme

- Sender requests the right to transmit. Receiver sends a credit allowing transmission based on buffer and processor capacity.

— Lost credits can leave both sides hanging, so timeout mechanisms are necessary.

COMMENTS ON FLOW CONTROL TECHNIQUES

— Hybrid forms combining the four layers of flow control may be warranted

— What is the interaction between levels of flow control?

— A conservative hop-level flow control algorithms will limit the need for complex schemes at higher levels.

— Although there are many options, flow control design is limited by preexisting protocol structure, limits of storage, and limits of processing resources.

— Flow control algorithms which are both efficient and prevent deadlocks and congestion are hard to implement and difficult to verify.

— Gateways may require a special algorithm of flow control since they may have to interconnect networks employing incompatible flow control techniques.

REFERENCES: FLOW CONTROL ALGORITHMS

1. Marlo Gerla and Leonard Kleinrock, "Flow Control: A Comparative Survey," IEEE Transactions on Communications, Vol. COM-28, No. 4, April, 1980.

2. Andrew S. Tanenbaum, Computer Networks, Prentice-Hall, Englewood Cliffs, New Jersey, 1981

2.1.3.4.6.6 Message Segmentation

The application programs will be allowed to request movement of large blocks of data, with no regard for the physical limitations imposed by the message transfer implementation. For example, the user may ask to read fifty thousand characters of text from a file, even though the network transfer medium is designed for maximum block lengths of one thousand characters. The host processor is also likely to have maximum message lengths which it can handle, such as an input/output limit of thirty thousand characters per block. The trade study must determine the sizes of packets to be transmitted. A number of packet sizes may be utilized to facilitate both small and large scale data transfers. The actual segmentation will be done in the hosts.

If fixed block sizes are utilized, an implementation issue is whether or not to allow multiple messages to be sent within one block. If only one message is allowed, there is a large potential for wasting bandwidth, such as a data item of a few bytes being sent in a 1KB block. If multiple messages are allowed, should all the messages be destined for one user? If there is a bit error during transmission, should all the messages be retransmitted? If not, what is the mechanism which points out the proper message to retransmit? An example is the common practice of sending an acknowledgment to a previous packet along with the next message packet. Once the packet arrives, the acknowledgement and message are treated independently.

2.1.3.4.7 DISTRIBUTION OF NETWORK-ORIENTED DOS TASKS AMONG NIUS AND HOSTS

Network oriented functions such as routing, flow control, and error control have to be divided among the NIUs and hosts in the network. A significant portion of the network protocol will be resident in the NIUs.

The distribution of tasks between the NIU and host can be readily determined by studying the OSI 7-layer protocol. In this system, network communications functions are addressed by layers 2-4. All higher layers (5-7) address user-oriented function such as security, assisting in remote logins, and file transfer packages. Since layers 5 and 7 are dependent upon the application, they are generally handled within the hosts.

Layer 2, or data link protocol, is responsible for media access, framing individual packets, handling transmission errors, and flow control. These functions are best suited for the NIUs. Options for data link flow control are described in section 2.1.3.4.6.5 Media access control and error control is addressed in a separate options paper (2.5.3.2).

Level 3, or NETWORK LAYER, is concerned with such issues as "What does the NIU/host interface look like?", routing within the network, and handling/preventing congestion and deadlocks in the network. Of these, congestion handling/prevention and routing schemes are better suited for implementation in the NIUs as these functions are dependent upon knowledge of network conditions. The implementation of the NIU/host interface, however, requires careful study.

Layer 4, or TRANSPORT LAYER is concerned with end-to-end connections between processes. This layer is responsible for the initial breakup of messages into segments. Within the header are segment number of the message and the source and destination processes. This layer may also include a final error check to ensure that packets were not garbled in transmission from the last NIU to the host.

Layer 5, can be thought of an interface between higher level presentation and application layers and the network communications functions handled in layers 2-4. Services such as maintaining a record of the progress of such communication (checkpointing) are performed by this layer.

Layer 6, or presentation layer, is concerned with packages for changing the format of the data. This may be done through encryption, character code conversion, or graphics format conversion, among others. Of the functions, security belongs in the host to preserve end-to-end security. The other two functions may be performed in either the host or the NIU.

Layer 7, or application layer, consists of a set of functions which may be frequently utilized by application programs. One such function is a file transfer protocol.

One possible method of dividing network communication functions between hosts and NIUs is provided below. Further discussions of this topic may be found in the Network Interface Devices (1.7.1.2) options paper. The Onboard Local Area Network Trade Study will further address the issue of task division between hosts and NIUs.

Network functions which may be performed in the hosts include:

- Initial breakup of messages into segments, including placing sequence numbers and src/dest. processes in the header.
- Addressing (optional)
- Error checking to ensure end-to-end error control
- End-to-End flow control
- Reassembly of segments into messages and duplicate control if not done in the NIUs
- Some layer 6 functions (may split with NIUs)

NIU tasks may include:

- All data link layer functions
  - Frame packets, media access, NIU-NIU flow control, error control
- All network layer functions
  - Packetization/Reassembly of packets
  - Routing, flow control, and congestion control
  - Duplicate control
- Transport layer functions (optional)
  - Converting from logical to physical addresses (Optionally in hosts)
- Some presentation layer (6) functions (e.g., encryption)

REFERENCES: IMPLEMENTATION IN NIU OR HOST?

1. J.H. Saltzer, Et.Al., "End-to-End Arguments in System Design," in Proc. 2nd Int. Conf. Distrib. Comput. Syst., April 1981

2. Andrew S. Tanenbaum, Computer Networks, Prentice-Hall, Englewood Cliffs, New Jersey, 1981

2.1.3.5 NETWORK PROTOCOLS - THE ISO/OSI 7-LAYER PROTOCOL

A number of varying protocols have been used for network communication, including, among others, the ARPANET protocol, IBM's SNA (System Network Architecture), and Digital Equipment Corporation's DECNET. The International Standards Organization's Open Systems Interconnection (ISO/OSI) reference model is intended as a first step towards standardization of the various protocols.

The ISO/OSI model, like others, divides the task of network communication into layers of abstraction. Each level is independent of the others and lower levels are completely transparent. With this type of model, the protocol governing host-host communication need not worry about network details such as routing and congestion control. This section will briefly present the 7 layers of the OSI reference model and lists the functions associated with each level. Figure 5 summarizes the interaction between layers. Note that the figure shows only layers 1-3 as being performed in the NIUs. This division, as indicated by the previous section, is a subject of much controversy and study. For further details, the interested reader may consult (1) or (2). (1) is a general text reference, (2) is oriented towards the requirements of Space Station.

COMMUNICATION NETWORK ARCHITECTURE (LAYERED PROTOCOL)

o ISO OSI 7 LAYER REFERENCE MODEL USED TO DEFINE PROTOCOL

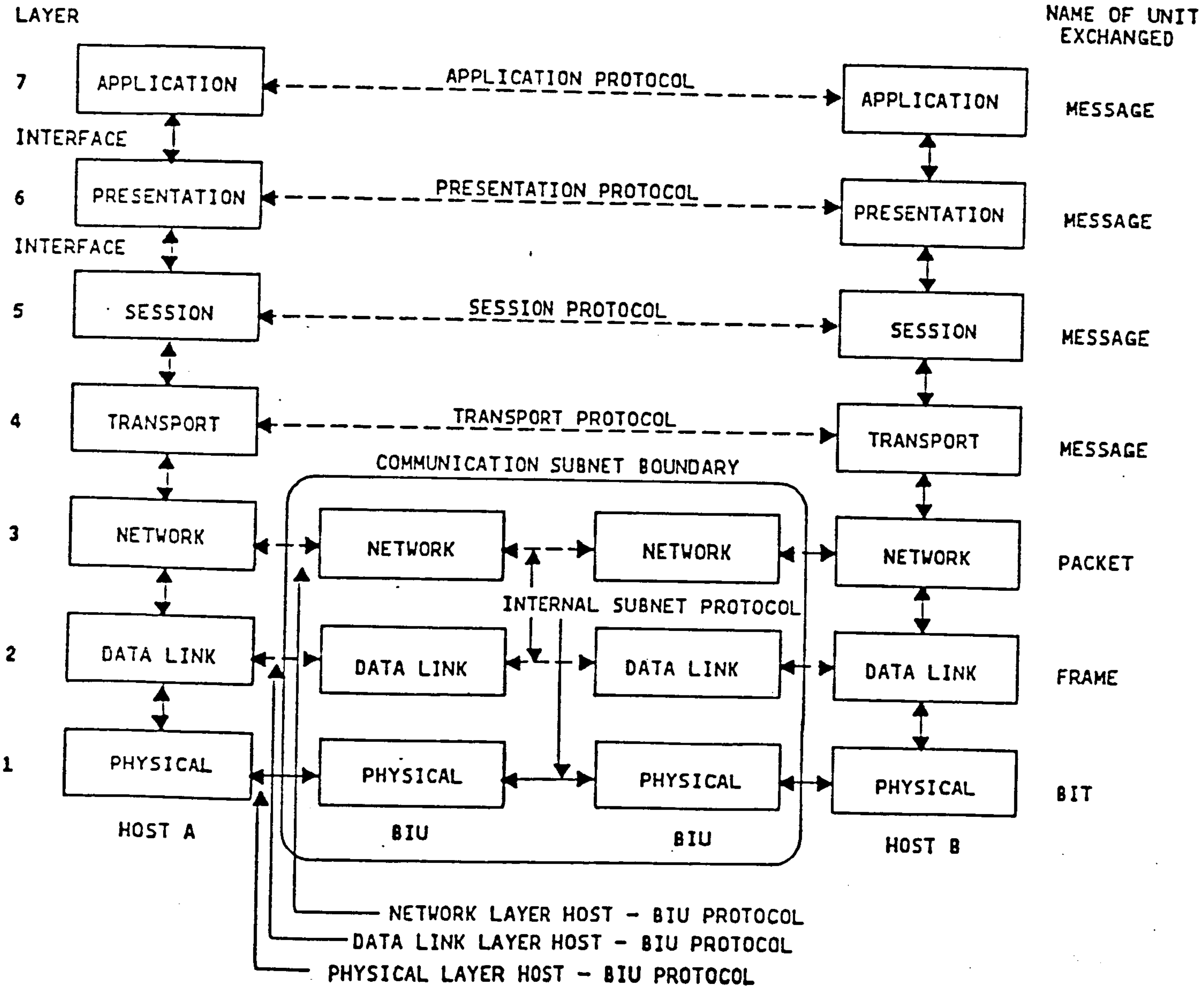


Figure 5. Interaction Between ISO/OSI Layers

Layer	Name	Function
1	Physical Layer	Transmission of data bits across physical media
2	Data Link Layer	Transmission of data blocks across a physical link. This link may connect two adjacent NIUs or connect a host and its associated NIU
3	Network Layer	Transmission of data blocks across the network
4	Transport Layer	End-to-End communication between two hosts
5	Session Layer	End-to-End communication between processes
6	Presentation Layer	Performs necessary syntax conversions between incompatible data formats and device access formats
7	Application Layer	Performs mechanics of information exchange on behalf of source/sink user application processe

The principle functions addressed by each layer are the following:

Data Link Layer (2) — Framing, Error Control, Media Access, Sequencing (optional), and Flow Control

Network Layer (3) — Routing/Switching/Relaying, Flow Control,
Sequencing (optional),
Error Control (optional), Financial Accounting
Type of service: Virtual Circuits or Datagrams
Address Mapping (optional)

Transport Layer (4) — End-to-End service: VCs or Datagrams?
Segmentation/Reassembly, Error Control,
Multiplexing Connections, Flow Control,
Sequencing (optional), Address Mapping

Session Layer (5) — Sequencing (optional), Checkpointing

Presentation Layer (6) — Data Encryption, Character Code Conversion,
Data Compression, Graphics Syntax Conversion,
Virtual Terminal Protocols

Application Layer (7) — Virtual file protocols,
Resource Access Transparency,
Interprocess Communication Mechanisms

REFERENCES – ISO/OSI REFERENCE MODEL

1. Andrew S. Tanenbaum, Computer Networks, Prentice-Hall, Englewood Cliffs, New Jersey, 1981

2. Computer Technology Associates, Inc., Computer Networking Study, Final Briefing, 1985

2.2 SYSTEM ARCHITECTURE

2.2.1 Fault Tolerance

System requirements for high reliability and availability are achieved by a combination of fault avoidance and fault tolerance. Fault avoidance reduces the possibility of hardware failures through conservative design procedures and use of high-reliability components, and the possibility of software failures by testing and verification procedures. Fault tolerance uses redundancy to recover from failures. The SSDS cannot meet availability requirements by fault avoidance alone (for example,physical damage must be considered), so that fault tolerance must be provided. This paper describes several classical techniques for fault tolerance, in four general steps. First, detection that a fault has occurred. Second, immediate treatment of the fault to minimize damage and begin recovery. Third, assessment of the problem. And fourth, full recovery and continuation of normal operations.

None of these techniques is applicable to the data management system as a whole, but may be applicable to individual parts such as processors, network interface units, interconnection media, etc. Several techniques rely on customized design, such as processors with internal redundancy, which have the potential for high maintenance costs because of their non-standard nature. Some techniques are useable for standard equipment. Also, in many instances, the SSDS will include commercial, military, or NASA off-the-shelf equipment for which the fault detection and recovery is already built-in and outside the control of the SSDS trades. For example, most ground processors will be commercial equipment; the on-board processors are likely to be a space qualified version of a large volume military or NASA processor; and both the ground and on-board data base management software are expected to be based on a commercial product instead of a new development. Individual applications must make the judgement of which methods are available and applicable to meet their requirements. In particular, ground SSDS requirements for fault tolerance are quite different from the on-board requirements, in view of the long resupply cycle, the possibility for EVA (extra-vehicular activity) for replacement, and the potential for unmanned operations.

Most functions of the core and payload systems will utilize their own internal methods for fault tolerance. Typical examples are cross correlation of different sensors in navigation of the space constellation, integrity checks within the data base management system, or leak detection of the reaction control system. This paper cannot cover the wide variety of special methods available to each SSDS function. The paper does address the methods for fault tolerance of the data processing and local area networks used to support those functions.

Current application needs for fault tolerance reflect five concerns:

1) Reliability. Reliability is essential in even the most basic computer applications. As the cost of the system decreases, the cost of maintenance and downtime grow to dominate system life-cycle costs, dovetailing with the users' need for less expensive maintenance approaches. System developers can offset the increasing cost of on-site service with systems that are more reliable and allow deferred repair.

2) Integrity. The system must be able to detect failures and, once a failure has occurred, restore the data to a consistent state before allowing computation to proceed.

3) Availability. As on-line access to information becomes more essential to an operation, long periods of downtime become less and less tolerable. The system must have the redundant resources to permit quick recovery. The allowable downtime varies, ranging from under a second for real-time control to several minutes or hours for less critical functions.

4) Degradation. Some applications require the computer equipment to operate in isolated locations (space vehicles, relay stations, etc.). For these applications, the computer system must continue functioning for very long periods of time, even if operation of the system is degraded.

5) Continuous cooperation. Because downtime can cause reduced productivity or even loss of life, continuous operation is often required. Downtime must be held to an absolute minimum in such applications.

2.2.1.1 Error Detection

The first requirement for fault tolerance is to be able to detect that a fault has occurred. Some techniques also recover from the fault at the same time, usually with status available to an external monitor.

2.2.1.1.1 Description

Several widely used methods to detect errors within a data processing system are:

- Built-in test equipment (BITE)
- Watchdog timers
- Parity
- Cyclic redundancy checks (CRC)
- Error correction codes (ECC)
- External checks

Fault detection may be performed either internal to a single unit or by use of an external monitor. Internal checks are sometimes called built-in test equipment (BITE), and is used by the manufacturer both for unit checkout and to satisfy specified levels of fault detection and isolation required for maintenance. The BITE may include watchdog timers to detect loss of critical functions (such as oscillator stopped or a dead interface unit). Many movements of data within a unit can be checked by simple parity or CRC to detect errors, or be enhanced to include error correction coding (ECC) to both detect and correct errors. Movement of data between units may use simple parity for error detection, or more complex cyclic redundancy checks (CRC), or ECC for both detection and correction of errors. External checks may include voting of the outputs of two or more units to detect and identify a faulty unit. Watchdog timers, parity, CRC, and ECC may also be used external to a single unit to detect problems with data transfers.

2.2.1.1.2 Option Characterization

BITE is usually active in a unit, providing a continuous detection capability. The coverage is often in the range of 80 to 95 percent of the failure rate of a unit, so that BITE by itself does not give the very high coverage (such as 99.99 percent) required for fault isolation and repair. BITE will typically be 10 percent of the total hardware in a unit.

Watchdog timers are usually used for coarse checks on interfaces between separate physical units, such as between a processor and an external device. Detection of a completely failed external device is very good, and prevents a processor from waiting forever for a failed device. Assessment of data quality (such as meaningless content or lost bits) is poor. Watchdog timers, either hardware or software, are usually an insignificant amount of a unit.

Parity, cyclic redundancy checks (CRC), and ECC are common methods of detecting errors in computer data transfers. Parity is the classic method to detect loss of a single bit of data, where the correlation of errors between bits is known (or presumed) to be negligible. Most computer memory devices include parity checks or better detection methods. CRC extends the parity concept to detection of multiple bits in error for cases where errors are likely to be grouped, such as serial transfers over a communication link. ECC is a further enhancement that uses redundancy within the information (extra bits) to detect and to correct errors. The parity/CRC/ECC type of detection provides very high detection of errors that fall within the designed types of errors. (For example, parity gives 100 percent detection of single bit errors, but only 50 percent detection of multiple bit errors.) Parity used in a memory adds one bit for every 8 or 16 bits (depending on the memory system). Parity used in a serial link uses one bit for every 8 or 16 bits transferred. CRC used in a serial link uses 8 or 16 bits for the entire message, and therefore uses only a small part of the bandwidth. ECC is being applied more widely as memory prices fall, with a common practice being to use 6 ECC bits for each 16 data bits in memory to correct any single bit error and detect any two bits in error. ECC on communication links, to cover long bursts of noise or data dropouts, can use much of the available bandwidth.

Voting and external checks provide good coverage of the failures for which the checks are designed. Coverage is virtually 100 percent if all outputs can be guaranteed to be identical with the equipment operating normally (a difficult goal to meet in practice). Coverage of non-identical outputs can range from very good for "sudden" failures, to poor for slowly diverging data. The general problem for non-identical outputs is to define tolerance levels that detect failed units without also rejecting good units that are near their specified performance limits.

2.2.1.1.3 Projected Capabilities

BITE and watchdog timers are classic methods to provide fault detection in the 80 to 95 percent range. No substantial improvement is expected during the Space Station timeframe.

The parity/CRC/ECC type of fault detection can be expected to evolve toward increased use of ECC to mask errors in memories as these devices increase in storage capacities and decrease in price/bit. Application of ECC such a Reed-Solomon encoding to serial communication links can be expected to increase as higher bandwidth channels evolve which can tolerate the loss of data bandwidth inherent in such encoding.

2.2.1.1.4 References

None.

2.2.1.2 Hardware Replication and Reconfiguration

This category includes options that represent broad classifications for the better known redundant hardware configurations (both static and dynamic). The options include the following:

- Standby Sparing
- Reconfigurable Duplexing
- Pair-and-Spare

- N-Modular Redundant (NMR) Voting
- Reconfigurable Voting
- Reconfigurable Multicomputers
- Reconfigurable Multiprocessors

2.2.1.2.1 Description. The most common redundancy strategy used to achieve high fault tolerance "coverage" is hardware replication. This can take the form of static redundancy where fixed configurations can tolerate failures (e.g., majority voting schemes) or dynamic redundancy where configurations can be dynamically changed to respond to faults or to allow online repair (replacement strategy). Static configurations are mature and have been used in many NASA and military applications. Dynamic redundancy represents an evolving technology that is closely tracking the evolution of distributed data processing. As a result, many distributed configurations have the potential to support reconfiguration and replacement (self-repair) strategies. In addition, some degree of static redundancy is often incorporated as part of an overall reconfiguration scheme.

Reconfiguration is triggered either by internal detection of faults in the damaged unit or by detection of errors in its output. Thus, fault detection techniques (defined in Section 2.2.1.1) will significantly influence a system's chance of successful reconfiguration. Other important factors include methods for fault confinement (limiting the impacts of a fault), damage assessment (identifying the extent of the effects of a fault), and resource switching to recover from the fault. After detection and reconfiguration (if necessary), the effects of errors must be eliminated through some form of recovery procedure (described in Section 2.2.1.4).

2.2.1.2.2 Option Characterization

a. Standby Sparing - This is a dual-redundant configuration based on a replacement strategy where a previously unused component is directly substituted for the failed primary (active) component. The replacement strategy can be manual, dynamic or spontaneous as defined below:

- Manual - The system takes no part in the replacement strategy (e.g., recabling, unit replacement, etc.)
- Dynamic - The system, responding to external stimuli, uses automated provisions for reorganizing future activities.
- Spontaneous - Strategy is carried out entirely by the system itself (self-repair) such as the JPL Self Test and Repair (STAR) system.

Standby replacement strategies can also be characterized by the degree of readiness of the backup component (i.e., time required to execute the replacement and recovery). Backup spares can be powered-down or inactive (cold backup), or executing redundant software (hot backup). The BELL ESS series and several transaction-oriented commercial systems (i.e., TANDEM, AURAGEN, etc.) use some form of checkpointing or synchronization to provide active backup capability. In the TANDEM NONSTOP system, process pairs execute in separate CPU's. The primary process handles all I/O and data base modification while the backup tracks all I/O messages at primary checkpoints and can take over "on-the-fly".

The key to implementing a dual-redundant replacement strategy is the ability to detect faults in the primary system. This can be accomplished by a variety of self-check techniques such as "I'm-alive" messages, timeouts, diagnostics, error detection codes, etc. (see Section 2.2.1.1). An alternative approach for active standby systems is to use hardware comparators to detect mismatches. In this case, both units are operating in parallel but only one is connected to the system output via a physical or logical switch. Once a fault is detected as the result of a comparator mismatch, the difficulty is to determine which unit is faulty (See 2.2.1.3).

The common techniques to identify the faulty unit are:

- Run self-test diagnostics (BELL ESS 2)
- Include additional self-checking circuitry. A joint occurrence of an internal fault and a comparator mismatch determines faulty unit.
- Watchdog timer (BELL ESS 2)
- External unit to compare results of known computations with stored constants, etc.

As an alternative, two pairs (quad redundancy) can be used (see pair-and-spare below) and a mismatch in the primary pair will result in automatic switching to the backup pair.

b. Reconfigurable Duplexing. This approach has been employed in many dual-processor systems where both units are used to enhance system throughput (i.e., non-redundant processes). The duality of these systems offers some features to support a level of reconfigurability in the case of processor failure. This, of course, results in degraded performance and generally requires significant time to reconfigure and cold start the system. In reality, these systems have many single-point failure components, do not support the removal or return of components without powering-down, and lack sufficient software support.

Reconfiguration limited to a duplex configuration will probably receive little attention in the future for fault tolerance. The extension of reconfiguration to multicomputer and multiprocessor systems is discussed later in this section.

c. Pair-and-Spare - The pair-and-spare concept employs two pairs of components (quad replication) with one pair functioning as the backup "spare". Each pair (often on a single board) is self-checking and each pair consists of identical functions that receive identical inputs. Output comparators generate an error signal whenever the outputs of a pair don't match. Normally, the two pairs are tightly synchronized and should one pair detect an internal mismatch, it merely "disconnects" and the spare continues to carry the load, all without missing a beat. This technique is employed in the Stratus computers, the AT&T 3B20D processors and Intel's 432 as well as several other in-development systems.

One of the major advantages of this approach is that no explicit recovery action is required since the process continues with the spare while the faulty unit is replaced. When a repaired unit is returned, an interrupt can be generated to re-establish synchronization. Since no other recovery is required, the system appears to the user to be a conventional computer requiring no special fault tolerant programming consideration.

This degree of redundancy has become practical (cost-effective) because of the advent of powerful microprocessors such as those described in Section 1.3. This is further enhanced by new microprocessor features, such as those provided by the Intel 432 family, which facilitate the construction of pair-and-spare and/or self-checking systems. These features are "built-into" the chips and can be activated by external signals. However, the approach is of limited value when physical damage must be tolerated, since the tight synchronization requires only a small spacial separation.

d. N-Modular Redundancy (NMR) Voting. This approach is one of the oldest techniques for achieving fault tolerance with fixed replication and majority voting. The most common example of this technique, Triple Modular Redundancy (TMR), involves three identical modules and majority voting circuits which check the module outputs for exact equality. It is thus designed to "mask" the failure of any single module, by accepting any output that at least two of the modules agree on. The voting mechanism can also include a means of identifying the faulty module and providing notification. TMR is also incorporated into more sophisticated reconfigurable strategies such as SIFT and FTMP, below.

Schemes such as TMR are based on the assumption that failures in the different modules are independent of each other. Thus, a design fault (hardware or software) may not be masked. In addition, it is necessary that the modules do not interact with each other. The Charles Stark Draper Laboratory (CSDL) has developed a fault-tolerant multiprocessor (FTMP) based on groups of TMR processors.

Most TMR voting systems use hardware voting circuits. A variation on this theme is the Software Implemented Fault Tolerance (SIFT) system developed by Standard Research Institute (SRI). As the name implies, the emphasis is placed on using software techniques to minimize the amount of special purpose hardware. This allows fault tolerant systems to be constructed using general-purpose modules and standardized networking capability. The SIFT approach has now been adopted in new commercial system (i.e. August Systems).

Reliability characterization of NMR static configurations is relatively well understood, and well-established predictive models exist.

Another approach to modular redundancy is illustrated by the Space Shuttle on-board computer system. During the critical ascent and entry phases, the system uses quad-redundancy (4 computers) for the primary processing, and a fifth computer with independently programmed backup software tracking all sensor input data and ready to take over vehicle control. The switch to the backup is manually commanded by the crew if they judge that the primary system has failed. Orbital operations normally use dual-redundancy for vehicle control, with a simplex computer used for system management and manipulator arm processing. From experience, the most important fault detection technique is a "fail to sync" of one of the computers of the redundant set, which has been caused by problems like a permanent computer failure, an internal register transient failure, a software timing error, or a fault in the input/output processor. The exchange and cross-comparison of critical data has almost never detected an error before a loss of sync. With three or more computers in the redundant set, the failed unit is easily identified by a majority vote. With one or two computers, the faulty unit is usually identified by the loss of updates to the display controlled from that computer. The crew is responsible for turning off the failed computer and reassigning any devices to the control of another computer. An early policy decision was that there would be no automatic reconfiguration of computers or reassignment of input/output control, because of the concern that a computer might fail in such a way as to enter that code by mistake and preempt control of the vehicle.

e. Reconfigurable Voting - This is a variation of the NMR static configurations where additional spares can be substituted for failed modules of the primary voting set. This requires a "disagreement detector" and a switching mechanism that can identify the faulty module, inhibit it from further participation, and enable an active spare module to be included in future voting. This hybrid approach can result in substantial improvements in reliability over standard NMR.

f. Reconfigurable Multicomputers. This classification includes a wide variety of loosely-coupled distributed configurations that have benefitted from recent advances in network communications and VLSI processing capabilities. The flexibility offered by standard networking schemes allows the system designer to easily adapt to meet the needs of growth and/or changing requirements. Such systems can be reconfigured for "graceful degradation" and/or can include additional on-line spares for replacement of faulty units. A key features of these configurations is the potential for cost-effectiveness since a large number of processors can be "backed up" by a relatively small number of spares. However, most existing systems still use multiple pairs (i.e. Tandem) or TMR Triads (SIFT, August Systems, etc.). In the more sophisticated concepts using multiple TMR Triads such as SIFT or FTMP (classified as a multiprocessor) a pool of spares can be used to replace failed triad members.

Representative systems include the following:

- SIFT
- TANDEM NON-STOP
- AUGUST SYSTEMS
- AIPS

As the configuration interconnections become more complex and the methods for fault detection, reconfiguration/self-repair, and system recovery become more sophisticated, our ability to predict (model) system performance becomes more uncertain. This is particularly true for system reliability where the straight-forward and well-understood combinatorial techniques are limited in value. Complex configurations require a more detailed modeling capability such as Markov and fault-free analysis techniques. While these techniques are employed in modern modeling codes (i.e., CARE III, AAIES, MARK 1, etc.) their ability to effectively model various network interconnection reliabilities is yet to be established.

g. Reconfigurable Multiprocessors - This approach has the same general attributes as those described above for loosely-coupled system except that processors share memory in a tightly-coupled configuration. Representative systems include the following:

- Fault Tolerant Multi-Processor (FTMP)
- C. mmp (Carnegie Mellon Univ.)
- Synapse N + 1
- Sequoia Systems
- Pluribas

A disadvantage to this approach is that the shared-memory can become a single point of failure and a memory fault can wipe out critical information. This generally requires special provisions in the memory controller and the interconnection logic to minimize these effects. A more serious disadvantage is the lack of substantial growth potential to increase throughput since the shared memory can often become a system "bottleneck" (unless cross-strapped, multi-ported memory is used). This limitation may not be important for some applications and multi-processors can be used as NMR nodes in an expandable loosely-coupled network (i.e., C.UMP, AIPS, etc.). One approach to alleviating this limitation is the use of high-speed cache memory within each processor to minimize memory/bus contention. This approach is used by the Synapse N+1 where one more processor than is required is configured as a spare.

2.2.1.2.3 Projected Capabilities

Networks of distributed processors are advancing rapidly, making the use of reconfigurable multicomputers practical in the 1987 and later time-frames. The network approach, including redundant communication paths among the processors, permits spatial separation for damage considerations, as well as growth by adding more processors to the network. This kind of system is emerging as the main processing element of the SSDS.

Two or more processors executing as a loosely coupled redundant set is an established technique for hardware, but does require that the software be able to reliably detect unacceptable differences between the outputs. This ability is highly dependent on the subsystem applications.

Tightly synchronized processors is a state-of-the-art technology that will advance by 1987. Because of its ability to detect and possibly recover from component faults, this approach may be useful for special single-board subsystem processors, especially where replacement may be difficult and continuous operation is necessary. The technique is unlikely to be applied to general processors or where physical damage of the entire unit is a design consideration.

2.2.1.2.4 References

None.

2.2.1.3 Damage Assessment

2.2.1.3.1 Description

Damage assessment will determine the degree of damage to the unit itself (as a start forward repair) and the effects on the system. This step may precede or follow actual error recovery, dependent on the particular application and failure. The classic techniques are self tests of a unit and correlation of trouble reports from multiple units or problem occurrences. A developing technique for ground processors is to control the diagnostics from a remote location (such as a phone link) so that the repairman can bring the required parts on a single service call.

2.2.1.3.2 Option Characteristics

2.2.1.3.2.1 Self Test

Fault detection will not necessarily isolate the failed unit. For example, a dual redundant set may indicate a disagreement, but show no explicit failure of either unit. Even for an explicit unit failure indication, repair activities will need a reasonable indication of the most likely failed internal components. A widely used technique to identify the failed unit, and the components within the unit,is to execute a self test. The basic requirement is that the unit is sufficiently good to execute a detailed diagnostic program. Therefore, the technique is not useful for failures of basic components like power supplies. The details of a self test depend strongly on the nature of the unit. For a processor, the approach is to perform a series of tests which build up confidence that the processor can execute every type of instruction (branch, add, logic, etc.), can read and write every memory location, can store various data values in memory, and can interface to all parts of its input/output section which do not actually activate external interfaces. If all these functions perform correctly, the self test can begin to transmit and receive data (for safety, usually after permission by an operator) to further check the unit's interfaces. Any failures are recorded. Perhaps 90 to 95 percent of the time, this self test will indicate both the failed unit and an idea of the possible failed component. The technique is very effective against solid failures, but does not do well in isolating transient failures or failures which depend on data values, sequences of operations, or timing.

If self test does not isolate the failed unit and/or component, it is probably necessary to remove all suspect units for return to the manufacturer for special analysis, such as stress tests of thermal cycles or vibration, using equipment not normally available at field sites. For the on-board system, the number of spares must consider the inability to always isolate the failed unit.

2.2.1.3.2.2 Trouble Reports Correlation

Transient failures are notoriously difficult to isolate, both because of their infrequent occurrence and because they often may be due to any of several units. Self tests, both in-place and at the manufacturer, are likely to show no problem, with the units returned for continued use. The usual method to eventually solve such problems (for those that do get solved) is to build up a correlation of trouble reports over time. The record keeping is critical to this effort. The units involved in each occurrence (including serial numbers), the time of occurrence, any unusual activities, etc. must be recorded. A pattern may eventually emerge to indicate some units or conditions that seem to induce a problem which suggests special tests that may force frequent enough failures to isolate the problem.

2.2.1.3.2.3 Remote Diagnostics

The use of remote diagnostics is developing as a cost effective method of assessing failures of a unit. Repair personnel use a phone link to start and control self tests or other diagnostics from a central location, with the intent of determining which parts will need to be carried to the site for replacement of the failed component. The technology for successful remote diagnostic capability applies to the SSDS ground processing system, and may also be applicable to the on-board processing system. Expert systems are evolving as a means to automatically analyze the diagnostic data to identify the most likely failed components, reducing the average down time of the system.

2.2.1.3.3 Projected Capabilities

Processors executing as a redundant set are an existing method of detecting faults in a system. The redundancy may be either at a low level (tightly synchronized) or at a high level (loosely synchronized, possibly tolerance checks instead of identical results). The decreasing cost of hardware relative to total system costs and the developing use of networks of processors should lead to increased use of redundant processors for fault isolation.

Remote diagnostics are emerging as a cost effective way to assess the cause of a fault for commercial equipment. The same technology which makes this a success can be expected to lead to improved diagnostics for the SSDS, including the on-board processors. The direct effect is that dual redundant processors, rather than triple redundant (for easy identification of the failed unit) may become practical even for critical on-board uses.

Fault diagnosis is one area that continues to be mentioned as an application of expert systems.. This technology is not likely to be useable in the 1987 time frame, but may develop by 1995.

2.2.1.3.4 References

None.

2.2.1.4 Error Recovery

Error recovery will resume normal operations after a fault. This step follows the fault detection and treatment to limit damage and start a replacement processor. Recovery may precede or follow damage assessment, depending on whether it is first necessary to determine, for example, which of a pair of processors has failed. Error recovery may be classed as either backward or forward. Backward error recovery anticipates errors by saving past states (such as checkpoints) or taking other actions before the occurrence of a fault. Forward error recovery reacts after an error has occurred, and does not rely on earlier actions. The preferred technique will depend on the particular application.

2.2.1.4.1 Description

The classic techniques are designed to protect primarily against hardware failures by providing a means to turn off a failed unit, turn on a replacement unit, go back to a previous point in time, validate the data at the point, and extrapolate to the current time. Protection against software errors includes the classic technique of independent backup software, and newer academic techniques of recovery blocks and N-version programming.

Software fault tolerance is defined as "a system's capability to perform its intended function in the presence of software faults". Most of the techniques in this category are still in the research stage and have seen little practical application. Current emphasis is placed on fault avoidance rather than tolerance, employing such techniques as structured design/programming, automated analysis/deign tools, and extensive testing/verification/validation.

A major issue associated with such techniques is that software reliability is difficult to model since software does not "fail"; it only contains residual design errors. While testing may detect most errors, it cannot guarantee the absence of all faults. As a result, software reliability predictions are typically based on the number of errors found during testing and the amount of testing; methods with high uncertainty at best.

2.2.1.4.1.1 Checkpoint/Rollback

This approach depends on the provision of recovery points for which the state of a process can be recorded and later reinstated. Such techniques have been used in many fault tolerant designs including those that are reconfigurable or self-repair (dynamic hardware redundancy), such as the JPL STAR, TANDEM's NONSTOP, Raytheon's FTSC, etc. The specifics of this technique vary depending on the hardware configuration and the amount of specialized hardware support for recovery. Fixed overhead is incurred to establish regular recovery points whether or not faults occur and variable overhead for rollback procedures when a fault is detected.

2.2.1.4.1.2 Audit Trail

Audit trails are used in conjunction with checkpoints for failure recovery in a number of applications. A typical application is in reconstructing critical data after a failure, such as banking transactions or data base updates. In these cases it is not adequate to simply go back to an earlier point in time (such as the bank accounts at yesterday's close of business), since mandatory changes may have occurred since that time. The audit trail is a record of all transactions since a checkpoint, at least up to the next checkpoint, that is adequate to supplement the checkpoint in reconstructing the current state. The minimum requirement is to record all changes in the audit trail before committing the change to the actual data base. Some applications may also require that access to data be recorded, as well as changes.

2.2.1.4.1.3 Information Validation

Most applications provide a degree of fault tolerance by validating information before its use. The validation may be part of recovery operations after a failure, or part of normal operations. Within the SSDS, a typical validation during recovery might be to check the integrity of a data base to make sure that the failure did not occur in the middle of an update, which could cause the data to be invalid. Normal operations frequently validate sensor readings for reasonable values, typically by comparison to an earlier reading of the same sensor or some average of a set of redundant sensors, both to detect failed sensors and to prevent action based on a fault reading. The particular method is highly dependent on the application.

2.2.1.4.1.4 Recovery Block

In the recovery block approach, there exists an ordered list of alternatives for a program, the first alternative being called the primary alternative. The alternatives are executed in order as required. Upon termination of an alternative, an acceptance test is performed to determine the "appropriateness" of the output. If successful, then the recovery block terminates with the output. If unsuccessful, then the next alternative in line is placed into execution after the local state has been reset. If all alternatives have been attempted without success, then the recovery block terminates with an error code.

Recovery blocks are an attempt to have the software protect itself against errors in itself. They are largely academic with little actual use. Some problems are the devising of acceptable alternative algorithms, the devising of acceptance tests, and the argument that adding the additional software to detect software errors just increases the chance of errors.

2.2.1.4.1.5 N-version Programming

N-version programming, like recovery blocks, has multiple copies, or versions, of a program. But unlike recovery blocks, there is no priority assigned to the different versions and all versions are executed upon a call to the program. (Execution may be sequential or parallel depending upon the computational resources available.) The final output of the program is determined by "vote" of the different versions. N-version programming can be integrated in a natural way with N-version hardware to provide for both hardware and software fault tolerance.

N-version programming is also an attempt to have the software protect itself against errors in the software, with the same problems as recovery blocks.

2.2.1.4.1.6 Backup Software

In the case of backup software, a separate backup system is maintained in the event that an emergency situation makes execution of the primary software untenable. Once control is passed to the backup software, the backup mode remains in effect until remedial action can be taken to correct the primary software. Typically, the backup system provides only those functions essential to safe operation.

2.2.1.4.1.7 Compensation

Recovery from some errors is done by compensation for the error condition, which may be incorrectable. This is a kind of reconfiguration, usually done within the software algorithms and data rather than by actual switching of equipment. A real example from Space Station would be the compensation for loss of a part of the attitude control system, such as a reaction jet used for reboost. The recovery requires compensation within the flight software by firing some other jets for a longer period of time to make up for the lost thrust. Checkpoints, backup software, or other techniques are not relevant to such failures. Within the data distribution of the SSDS, the message routing software might compensate for a failed or degraded link by selecting an alternate route, without actual reconfiguration or switching of any equipment. The form of compensation depends on the particular application and the anticipated failure modes.

2.2.1.4.2 Option Characteristics

One or more of these approaches can be applied in most situations to meet the needs of particular applications. The most common technique is to use checkpoints at a high enough frequency to permit fall-back to a previous point in time, followed by extrapolation to the current time (possibly by the use of an audit trail since the checkpoint). The generation of checkpoints may be initiated manually, or by the application, or by some automatic service within an operating system. For example, an operator may checkpoint customer records at the end of each business day, or Space Station navigation may checkpoint its state every ten minutes, or a data base management system may generate a checkpoint and related audit trail every hundred updates. Each application must determine the appropriate frequency and data to be saved. Validation of information is nearly always required, to cover cases of slow degradation that may have polluted one or more of the previous checkpoints.

Recovery block and N-version programming have limitations. In terms of efficient use of computational resources, recovery blocks have a decided advantage over N-version programming since the recovery block executes an alternative version of a program only when a fault is detected. With N-version programming, however, all versions are executed. There is little difference between the two approaches in terms of storage requirements. There is another criterion, in addition to efficiency, that distinguishes the two approaches. In the recovery-block approach the output of a program is subjected only to a generalized acceptance test, a test which in most cases will not embody total correctness. In N-version programming, on the other hand, the outputs of multiple versions of a program are compared, which really amounts to a check on the total correctness of a computation. The N-version programming and recovery-block techniques have seen relatively little use to date.

Backup software has a high tolerance of faults in certain applications, notably those where the backup software can take over with no prior dependence on the suspect primary system. A typical application might be for safing procedures in a process control environment. Somewhat less tolerance exists if the backup software must rely on the primary software to some extent. The

Space Shuttle backup software is typical of this type, since it gathers most of its sensor data while the primary software is in control by "listening" to sensor data which was requested by the primary software, and even uses some data computed by the primary software. Little fault tolerance exists if the backup software is highly dependent on the primary software, such as a data base management system for which the failure of the primary may have scrambled the data base.

2.2.1.4.3 Projected Capabilities

The classic techniques will continue to dominate error recovery. These are validation of information (sensors, etc.) before its use in computations, and checkpoints (including audit trails where appropriate) containing adequate data for resuming normal operations after a failure. Backup software can be used in very critical applications (safing of dangerous conditions) where there is a significant concern that the primary software may contain residual errors in critical processing even after extensive verification. Note that backup software can nearly double costs where it is used (two development, verification, and maintenance efforts), and should therefore be considered only for critical applications.

Recovery blocks and N-version programming are likely to remain as topics of theoretical software development, and find little use in practice. This opinion is based on the difficulty of finding multiple equivalent algorithms for complex applications, of defining adequate acceptance criteria to select the "correct" algorithm or version, and of assuring that the additional software does not itself contribute more errors than are being avoided.

2.2.1.4.4 References

1) AIPS Technology Survey Report, CSDL-C-5691, February 1984, by The Charles Stark Draper Laboratory, Inc., Contract NAS9-16023.

2) Workshop on Applied Fault Tolerant Computing for Aerospace Systems, Fort Worth, Texas, 8-10 November 1982, sponsored by NASA OAST and AIAA.

2.2.2 Autonomy/Automation

The application of autonomy and automation to Space Station is a topic of considerable interest and controversy. The two primary issues are 1) to what extent can the Space Station be made independent of a "cast of thousands" on the ground, and 2) what functions can be automated by machine without requiring interaction by ground or flight crew. Numerous studies have already been performed on this subject and most of the results from these studies tend to favor a maximal amount of function automation and space autonomy. The usual considerations driving these results being that a fully automated space station provides for maximal onboard productivity and a highly automated space station can be autonomous thereby requiring only a few ground crew people. These studies have, for the most part, not considered the cost-effectivity of implementing maximal space automation/autonomy.

Factors other than cost also affect the implementation of automation and autonomy. From an operation perspective, studies performed on the way people control complex systems in circumstances analogous to a space station (e.g., polar research stations, offshore oil drilling rigs, submarines) indicate that the onboard crew should have the prime responsibility to run the space station with the assistance of automation where appropriate. For example, if a human operator has to follow or track the details of what an automated system is doing (to guard against automation hardware failures) then automation has not decreased the operator's workload but may have increased his training requirements.

Preliminary results of the Advanced Technology Advisory Committee (ATAC) have been used in the preparation of this option section. The product of the ATAC committee will be a set of recommendations and automation and robotics options for use by contractors in their phase B space station definitions and preliminary designs. The final ATAC report will be issued in April 1985. The ATAC report, among other things, will assess the impact of various automation options concepts for use in space station. This assessment, performed by SRI International, determines the level of automation that would be technically feasible about 10 years after the initial

PRECEDING PAGE BLANK NOT FILMED

operational capability (IOC) is established. In addition, their task requires the identification of feasible levels of automation for IOC and design features required at IOC ("hooks" and "scars") to enable the integration of enhanced automation capabilities by future upgrades of hardware and software units.

The following definitions have been abstracted from various NASA and other documents:

Subsystem. First-level components of Space Station elements that can include, for example, the ECLS Subsystem, the Communication Subsystem, the Power Subsystem, the Propulsion Subsystem, etc.

Space Autonomy. The independence of the onboard subsystems from direct, real-time control by the ground (crew or machines) for a specified period of time.

Subsystem Autonomy. The independence of a subsystem from direct real-time control by external man or machine (e.g., an ECLS automated subsystem with its built in, fault-tolerant controller).

Automatic or Automated Process. A process that is controlled in an open- or closed-loop manner to achieve a goal or to minimize (maximize) a cost function specified for the process. The controller may be iterative and continuous with time or it may enter a terminal state (halt) once its goal is attained.

Conventional or Algorithmic Automation. An automation approach whereby a machine is programmed to respond to a predefined set of conditions with a predefined set of actions. The actions may be conditional, by use "IF-THEN-ELSE" statements, however, responses are governed completely by the designer's ability to anticipate the situations that the machine will face. Therefore, algorithmic autonomy works best for well understood situations.

Artificial Intelligence (AI). A branch of computer science dedicated to the design and implementation of computer programs which make human-like decisions, and can be adaptive and become more proficient at making decisions. AI systems interact with their operators in a "natural" way that mimics intelligent behavior.

Expert System. A component of AI, an expert or knowledge-based system is one that stores, processes, and utilizes a large data base of information about a specific area of knowledge to solve problems or answer questions pertaining to that area. The performance of the system is at the level of an experienced designer, teacher, technician, ..., who is specializing in this specific area of knowledge. Current expert systems store their expertise in a data base that has been derived from a human expert via an intermediate knowledge engineer. These systems are not self-adaptive or self-learning, but do have the ability to generate new concepts and relationships about knowledge already in the data base. Expert systems are candidates for implementing automation schemes.

Teleoperation ("Remote Operation"): Use of remotely controlled sensors and actuators allowing a human to operate equipment even though the human presence is removed from the work site. Refers to controlling the motion of a complex piece of equipment such as a mechanical arm, rather than simply turning a device on or off from a distance. Teleoperation requires that the human operator be able to see the object being manipulated, either directly or by visual sensors.

Telepresence ("Remote Presence"): The ability to transer a human's sensory perceptions, e.g., visual, tactile, to a remote site for the purpose of improved teleoperation performance. At the worksite, the manipulators have the dexterity to allow the operator to perform normal human functions. At the control station, the operator receives sufficient quantity and quality of sensory feedback to provide a feeling of actual presence at the worksite.

Augmented Teleoperator: A teleoperator with sensing and computation capability that can carry out portions of a desired operation without requiring detailed operator control. The terms "teleautomation" and "tele-robotics" have been used here.

Robot: A generic term, connoting many of the following ideas: A mechanism capable of manipulation of objects and/or movement having enough internal control, sensing, and computer analysis so as to carry out a more or less sophisticated task. The term usually connotes a certain degree of autonomy, and an ability to react appropriately to changing conditions in its environment. Robotics is a specialized discipline within the broader fields of autonomy and automation.

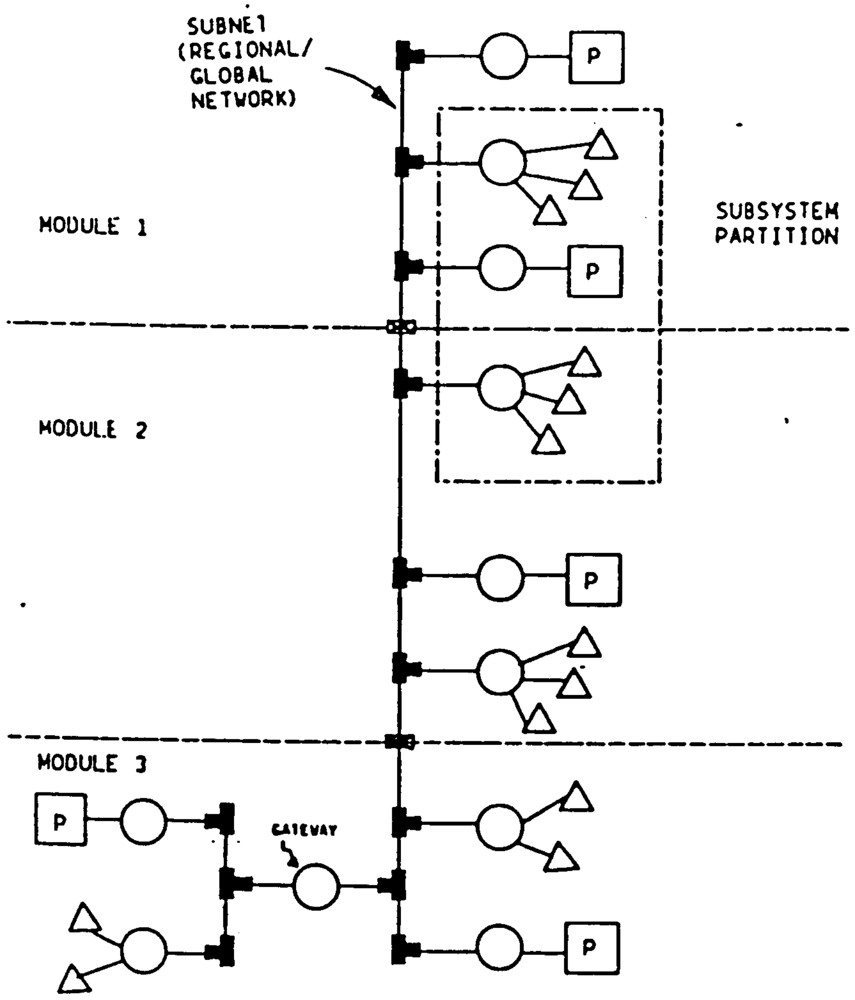
2.2.2.1 Subsystem Autonomy

2.2.2.1.1 Description

There are at least two possible distribution options that can be used, each of which can be considered to be of opposite extremes of a continuum of possible options: centralizated (minimum system hardware) and distributed (maximum subsystem autonomy). The "minimum hardware" option (Figure 1) groups all critical subsystems and accommodates their fault-tolerant needs by assigning them to replicated hardware elements. Noncritical subsystems are grouped and assigned to hardware elements that need not be replicated. In the maximum autonomy or "dedicated subsystem processing" approach (Figure 2), subsystem functions are allocated to dedicated processors, and fault tolerance is accommodated by replicating hardware elements. Processor types are selected from the prioritized options on the basis of subsystem resource requirements expanded with design margins, as above. This selection may require several standard processors to be used, depending on the dispersion of the subsystem resource requirements. Noncritical subsystems are grouped and assigned to one or more of these standard processors.

2.2.2.1.2 Characterization

There are numerous advantages and disadvantages that can be cited for centralized and distributed computer systems and these have been discussed in many places (see, for example, Space Station Program Description Document, Book No. 6, Appendix B, Systems Operations, Trade Study entitled "Centralized Versus Distributed Computer Systems", p.p. B-II-197-B-II-212, 8-1-83). A summary of the advantages and disadvantages of these approaches is given in Table 1.



MULTI-PROCESSOR HIERARCHY

∞ SOFTWARE INTENSIVE SOLUTION

∞ ANY COMPUTER CAN COMMUNICATE WITH ANY DEVICE

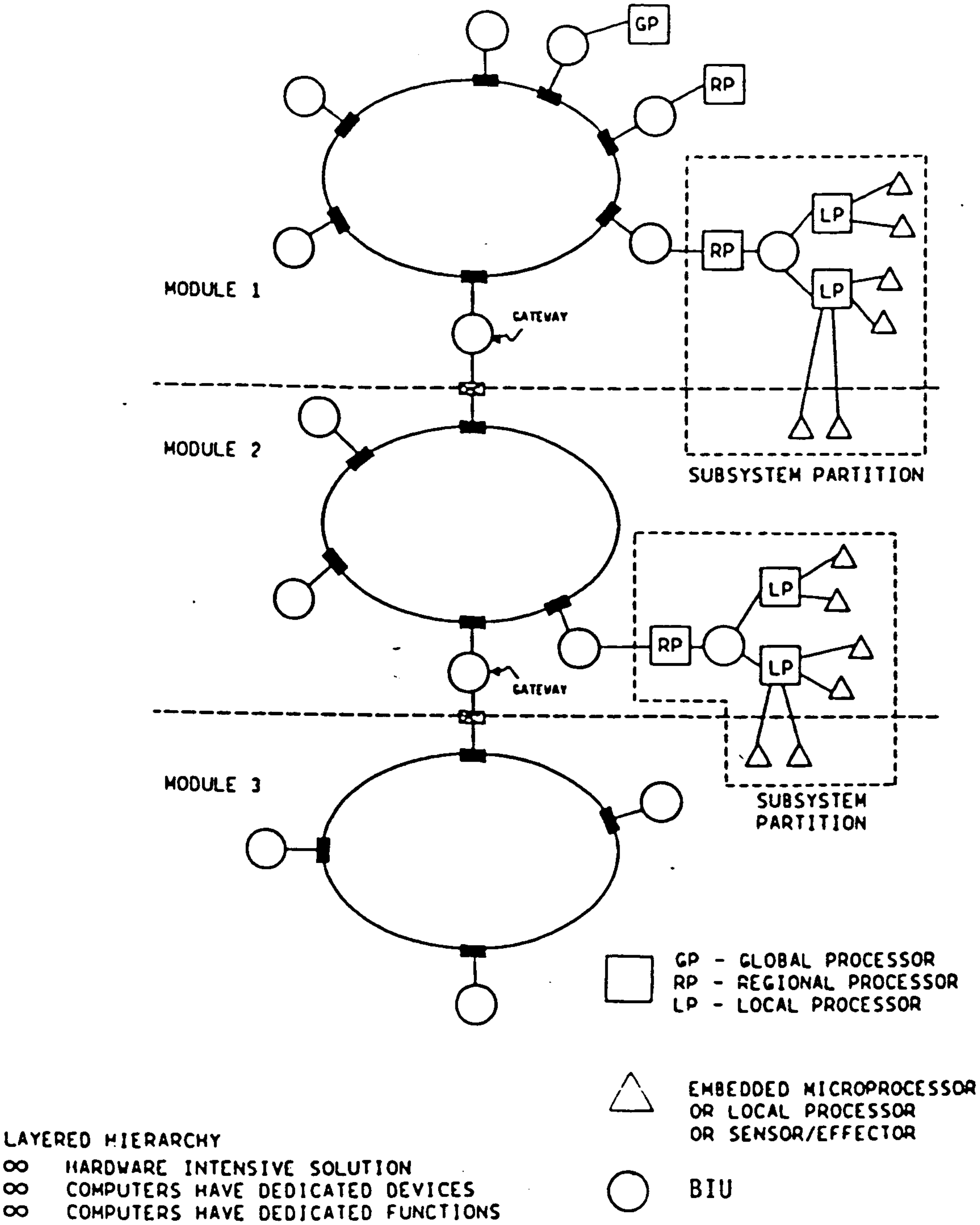
∞ COMPUTERS HAVE MULTIPLE FUNCTIONS

P PROCESSOR

△ EMBEDDED MICROPROCESSOR
OR LOCAL PROCESSOR
OR SENSOR/EFFECTOR

○ BIU

Figure 1. Centralized Architecture (Example)



LAYERED HIERARCHY

- HARDWARE INTENSIVE SOLUTION
- COMPUTERS HAVE DEDICATED DEVICES
- COMPUTERS HAVE DEDICATED FUNCTIONS

Figure 2. Distributed Architecture (Example)

Table 1

PROS AND CONS OF CENTRALIZED AND DISTRIBUTED ARCHITECTURES

	PROS	CONS
CENTRALIZED	o MINIMUM HARDWARE o CONCEPTS WELL DEVELOPED o FLEXIBILITY FOR GROWTH	o SOFTWARE INTENSIVE o COMPLEX INTEGRATION o DANGER OF CATASTROPHIC SYSTEM FAILURE IF PROCESSORS PHYSICALLY CENTRALIZED
DISTRIBUTED	o MAXIMUM SUBSYSTEM AUTONOMY o LOWER INTEGRATION AND TESTING COSTS o EASIER MAINTENANCE o SIMPLER SOFTWARE o FASTER DESIGN AND DEVELOPMENT TIME	o HARDWARE INTENSIVE o EXPENSIVE FAULT TOLERANCE o ADDITIONAL PROCESSORS FOR GROWTH

2.2.2.2 Function Automation Options

2.2.2.2.1 Description

It is possible, in principle, to automate all or nearly all SSDS functions, the degree of automation being limited by considerations of cost, schedule, value of human involvement, and other factors. Although a procedure may be automated, good human engineering practice requires that man must have the capability to issue commands individually in a manual mode, bypassing the automatic procedure, if desired. Therefore, he must be provided knowledge of what tasks should be done to be compatible with tasks in progress or completed. When one or more steps might be bypassed, each step must be monitored as it is performed and only bypassed by deliberate operator action. Exceptions to this rule occur only when computations are so complex , action is so precise, and the results of human error are so drastic that control must be removed from man and relegated to the computer.

A summary list of the SSDS functions is shown in Figure 3. The SSDS must provide SSDS resources for the implementation of each of these functions and the type and extent of these resources will depend on the allocation and degree-of-automation decisions (trade studies) associated with each function. It should be noted that the DECISIONS column in Figure 3 is blank since only the options are considered in this report; subsequent trade studies, selecting from options outlined here, will provide entries for this column. This paragraph addresses automation options and a subsequent paragraph discusses autonomy options.

2.2.2.2.2 Characterization

Automation techniques/options include those defined earlier in this report: 1) conventional or algorithmic, 2) artificial intelligence approaches, including the use of expert systems, 3) teleoperated and telepresence systems, and 4) robotics. Options for the use of artificial intelligence in automating functions are discussed in section 1.6 of the options report and all four of the above automation technologies are assessed in the ATAC report mentioned earlier.

	DECISIONS	
TOP LEVEL FUNCTION NAME	ALLOCATION	AUTOMATION
1.0 MANAGE CUSTOMER/OPERATOR DELIVERED DATA ° Onboard Data Capture (Real-Time/Delayed) ° Ground Distribution		
2.0 MANAGE CUSTOMER/OPERATOR SUPPLIED DATA ° Commands/Data, Address, User/Operator Validation ° Payload Command Res./Constr. Chk. ° Provide Ancillary Data ° Payload Data Processing/Operations ° OMV/OTV Operations		
3.0 SCHEDULE AND EXECUTE OPERATIONS ° Typical Day Payload Scheduling ° Short Term (2-7 Day) Schedule ° Prepare Time-Tagged Execution Sequence ° Execute Above Sequence		
4.0 OPERATE CORE SYSTEMS ° G&N, Att., Cont., Traffic Control, Tracking, Time/Freq. Management ° Power, Thermal Cont., Structures and Mech. Support, ECLSS ° Support Flight Crew Activities (Health, Safety, Habitability, EVA Support, Operations & Procedure Support) ° Provide Customer Avionics Services ° Caution & Warning, System Status, ... ° Diagnostics & Maintenance		
5.0 MANAGE FACILITIES AND RESOURCES ° Onboard Facilities ° Ground Facilities (SSCC, SAT, ROC, DHC)		
6.0 SIMULATE, INTEGRATE, AND TRAIN		
7.0 SUPPORT SPACE STATION PROGRAM		

Figure 3. Summary of SSDS Functions

Each automation option requires crew and SSDS resources (hardware/software) to an extent that depends on the degree-of-automation implemented by the automation option selected for each SSDS function (via the automation trade studies). The concept of degree of automation can be depicted graphically as shown in Figure 4. In this figure the horizontal line can be interpreted as a level or degree of automation with the left-most point corresponding to the labor intensive extreme (maximal crew resources) and the right-most point corresponding to the automated extreme (no crew resources).

The SSDS function set (Figure 4) is shown as being mapped into two regions and a point on this line which are termed Manual (M), Interactive (I), and Automatic (A), respectively. Another region to the left of M is where functions which require no SSDS resources (software/hardware) are mapped and these functions are not considered in the SSDS sutdy. An example of a non-SSDS function implementation would be the use of the U.S. Postal Service for transporting data on magnetic tapes.

The left-most point in the M region corresponds to the first point of interest to the data system and represents the automation level requiring minimal SSDS resources. As one proceeds from this point to the right increasing SSDS resources are required but with a corresponding decrease in crew/operator resources. At the automated extreme no crew/operator resources are required and the function is completely automated with the use of SSDS resources.

In summary, the option for implementing functions are A, I, and M. Further characterization of these options and the crew/operator/ machine allocation responsibilities is shown in Figure 5.

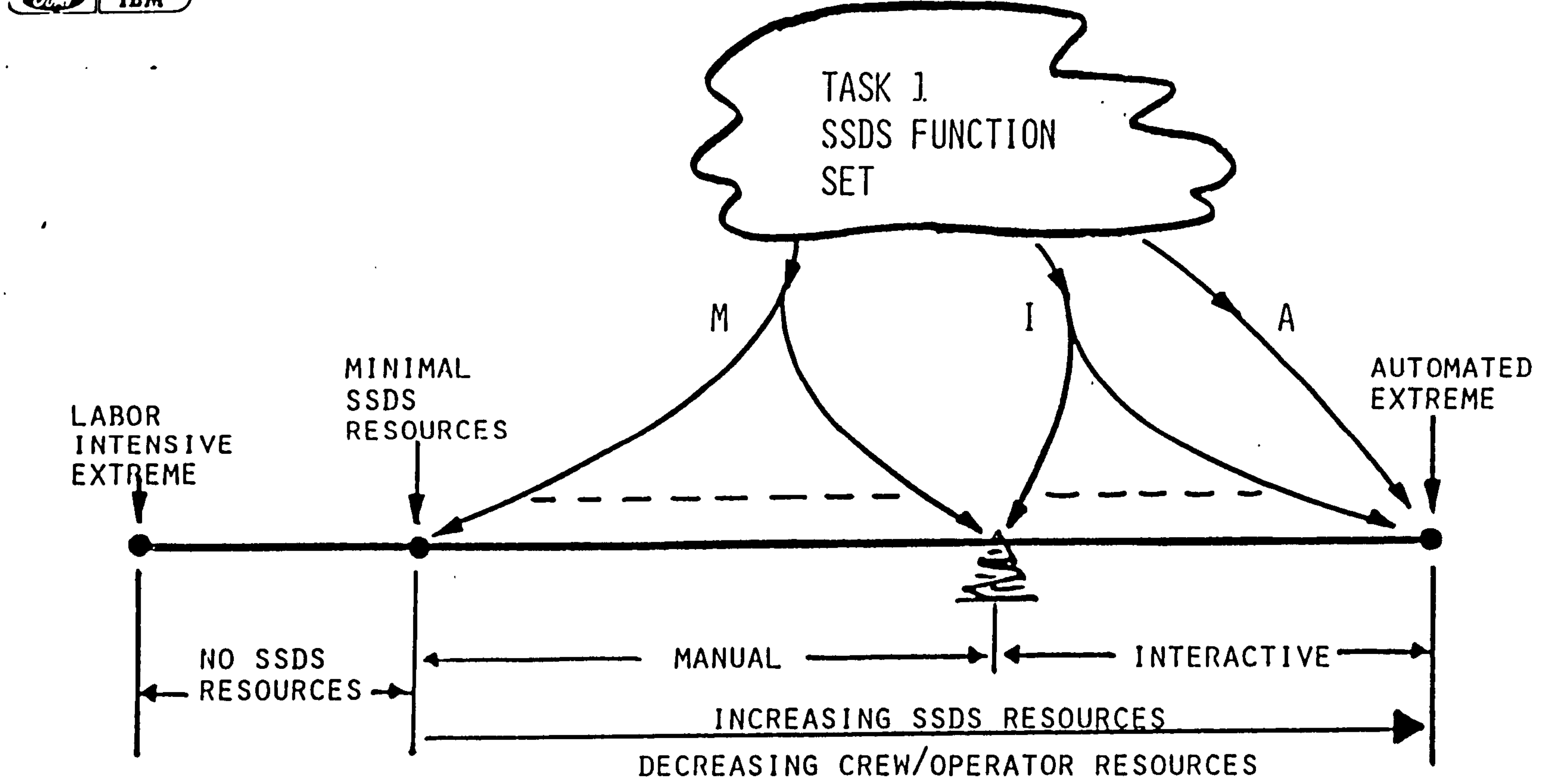
2.2.2.3 Space Station Autonomy (Ground/Space)

2.2.2.3.1 Description

The degrees of Space Station autonomy and the degree of subsystem autonomy are separable issues. Space Station autonomy implies an onboard capability to perform all essential functions, many of which have traditionally been done on the ground, usually by large numbers of people. To do these functions onboard

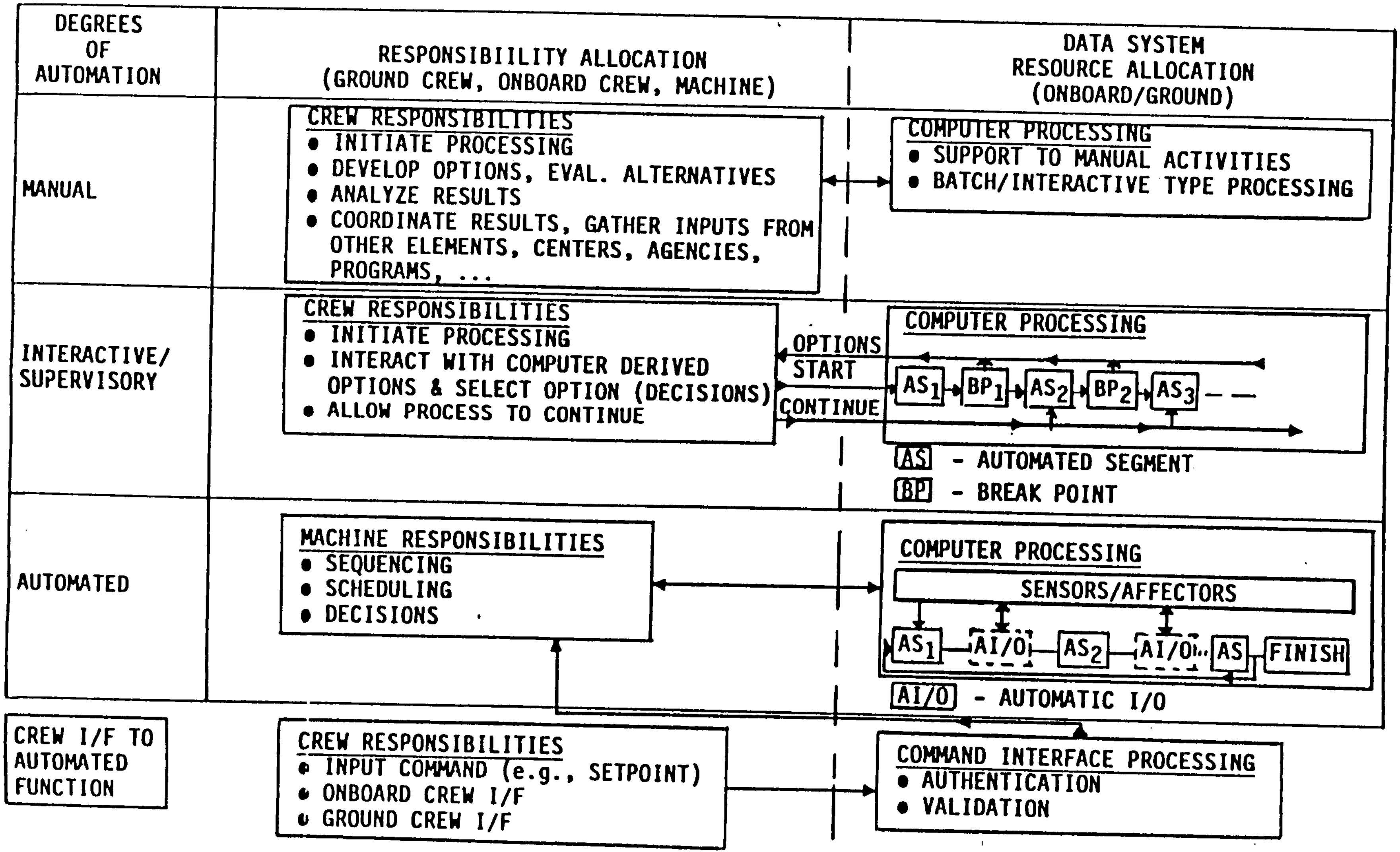
Figure 4

DEGREES OF AUTOMATION



SSDS MDAC Ford IBM

AUTOMATION AND THE CONCEPT OF RESOURCE AND RESPONSIBILITY ALLOCATION



DEGREES OF AUTOMATION

RESPONSIBIILITY ALLOCATION (GROUND CREW, ONBOARD CREW, MACHINE)

DATA SYSTEM RESOURCE ALLOCATION (ONBOARD/GROUND)

MANUAL

CREW RESPONSIBILITIES
- INITIATE PROCESSING
- DEVELOP OPTIONS, EVAL. ALTERNATIVES
- ANALYZE RESULTS
- COORDINATE RESULTS, GATHER INPUTS FROM OTHER ELEMENTS, CENTERS, AGENCIES, PROGRAMS, ...

COMPUTER PROCESSING
- SUPPORT TO MANUAL ACTIVITIES
- BATCH/INTERACTIVE TYPE PROCESSING

INTERACTIVE/ SUPERVISORY

CREW RESPONSIBILITIES
- INITIATE PROCESSING
- INTERACT WITH COMPUTER DERIVED OPTIONS & SELECT OPTION (DECISIONS)
- ALLOW PROCESS TO CONTINUE

OPTIONS
START
CONTINUE

COMPUTER PROCESSING

AS - AUTOMATED SEGMENT

BP - BREAK POINT

AUTOMATED

MACHINE RESPONSIBILITIES
- SEQUENCING
- SCHEDULING
- DECISIONS

COMPUTER PROCESSING

AI/O - AUTOMATIC I/O

CREW I/F TO AUTOMATED FUNCTION

CREW RESPONSIBILITIES
- INPUT COMMAND (e.g., SETPOINT)
- ONBOARD CREW I/F
- GROUND CREW I/F

COMMAND INTERFACE PROCESSING
- AUTHENTICATION
- VALIDATION

Figure 5

with few people requires a high degree of automation. The degree of autonomy depends on what level of automation is achievable and affordable as discussed earlier.

2.2.2.3.2 Characterization

The SSDS function to be allocated to space or ground were discussed earlier and are summarized in Figure 3. Many factors affect this allocation including: degree of automation achievable (non-automated functions consume valuable space crew time), cost of space resources (significantly higher than ground resources), communication bandwidth and availability requirements, life cycle cost, growth (once implemented on ground is it reasonable to re-implement for space migration?), safety, unmanned consideration, mission buildup (dual implementations for ground support during buildup), and so forth. Other than cost, the key drivers affecting allocation of function to onboard or ground are shown below:

1. Criticality - Allowable recovery time following failure of function implementation hardware.

2. Impact - Consequences of not meeting specified function criticality: loss of life, hazard/damage to SS, mission equipment, loss of key data, ...

3. Function Data Source and Function User (Crew/Customer) Physically Co-Located - Consideration is to implement function where input data is generated.

4. Space/Ground Communication Link Availability - Includes components of blind spot in TDRSS coverage, link MTBF and link MTTR.

5. Function Autonomy - Group functions with significant inter-function I/O rates and allocate as a group.

6. Response Time – Relates to transport delay encountered in space/terrestrial communication links (i.e., roundtrip TDRSS/DOMSAT delay ¨ 2 secs).

7. Space/Ground Communication Link Bandwidth – Considerations due to finite bandwidth allocated to SS.

2.2.3 System Growth

2.2.3.1 Description. One of the key requirements imposed on the SSDS is to provide sufficient growth capability to support the future needs of the program in a cost-effective manner. This includes growth in functional capability, available internal resources, and the ability to handle increased external interface demands. While applicable to both ground and space elements, the primary focus is on the more constrained environment of space where improvements and new capabilities may be accommodated by modification/replacement of existing modules or the addition of new ones. In any case, it must be recognized that future data system needs over the expected life-time of the space station cannot be accurately forecast and it will be difficult to determine "how much" growth potential is adequate. Therefore, SSDS architectural concepts need to consider multiple mechanisms for accommodating growth.

The traditional viewpoint of the space station evolutionary process is that phased development, new customers and improvement needs will result in data system requirements for more increased functional capability and/or resources. This section will describe the options for extending the SSDS capability to meet these requirements. An alternative viewpoint from the customer perspective, however, is that "growth" can be measured in terms of increased information derived from mission data. This can, of course, have the same effect as the traditional viewpoint if the result is increased data rates requiring additional SSDS functional/resource capabilities. The "information content" of the mission data handled by the SSDS could also be improved through additional onboard payload processing and/or data compression without necessarily increasing data rates. This could be particularly beneficial if applied to the relatively small number of high data rate missions. It is not likely that such payload unique functions would be included in the SSDS, however, they must be considered in a well-balanced, SSDS evolutionary process as a viable alternative to SSDS expansion.

The system SSDS growth options in this category include the following:

- Added/Modified Software
- Resource Margin
- Technology Insertion
- Resource Addition
- Network Addition

2.2.3.2 Option Characterization. The following paragraphs briefly describe the advantages, disadvantages and limitations associated with SSDS growth options. A summary is provided in Table I.

a. Added/Modified Software. The programmable nature of software suggests that the addition/modification of software is the "easiest" way of enhancing functional capability. Historically, this has resulted in two major problem areas. First, sufficient resource margin is usually not provided and as the resource limitations are invariably approached, attempts to optimize code or operate within these constraints generally result in much higher development costs and schedule extensions. Second, changes and enhancements to existing software often introduce new problems unless extensively controlled and rigorously tested/verified which will substantially increase life-cycle costs. Both problem areas are further compounded for large, complex, real-time applications such as the SSDS. These cost impacts can potentially be mitigated by modern software engineering techniques, modular design, and a "resource-rich" environment.

b. Resource Margin. This approach implies the selection of an SSDS "design point" that provides substantially more resource capability than is initially required to meet performance requirements. This "resource-rich" environment, when applied in a balanced manner (CPU, memory, bus, etc.), could provide sufficient margin to accommodate transient conditions incurred during initial buildup and until operational stabilization is achieved. This approach would also minimize the need to change the initial hardware configuration for some period of time resulting in a more stable environment. The extent to which the SSDS should be over designed will depend largely on physical constraints, funding profiles and a comparison of initial v.s. projected technology capabilities (technology insertion). A limiting factor may be the desire to minimize built-in technology obsolescence. However, the SSDS resource that may be most difficult to upgrade is the network interconnection structure. Careful attention must therefore be given to initial network bandwidth margin regardless of growth options adopted for other resources.

c. Technology Insertion. This option provides improvements in performance to weight/volume/power ratios in that existing "black box" can be replaced by higher performance, higher capacity components with minimal impact to system hardware or software interfaces. This approach requires strict adherence to a program of standardization that is developed early and incorporated into the architectural design of the SSDS. This program will address network interface standards, a standard HOL, and a set of standard ISA's for computers. However, standards must be selected that promote technology insertion, not inhibit it, i.e., adopted standards must be on a clear evolution path supported by industry and/or DoD. If radiation hardened components are required, then DoD standards must be considered carefully. In brief, the standards selected must be divorced from technology and supported by the entire technical community.

d. Resource Addition. This approach is similar in nature to that of Technology Insertion except that components are only added and these additions need not be a technology upgrade. An example of this option is the addition of more processors to handle new software requirements. If properly planned for in the architecture design, this approach can provide major improvements in functional and/or resource capability. The impact to the existing HW/SW configuration will depend on its functional interface with the rest of the system. An attractive way to minimize the impact of added resources is to use a network operating system (NOS) that provides some level of "virtual resource" capability making changes relatively transparent to the application software. This can be accomplished by either static or dynamic functional allocation processes and appropriate changes to the NOS data base. This is typically done via "process construction" and could support the reassignment of existing functions as well as the allocation of new ones. Again, this requires a modular design structure and interface standardization. The major limitation to adding resources (other than physical constraints) is generally the capability of the system interconnection structure. For example, typical local area network structures are often limited by the number of HW interconnections (HW addressing), distance between nodes and the network bus bandwidth.

e. Network Addition. This approach allows for the addition of networks to the SSDS as needed. One advantage of this technique is that new networks can utilize new technology. As long as the capacity of the existing networks is not exceeded, networks could be easily installed in new modules and then connected to the existing networks via a bridge or gateway. Minor software changes (e.g., table changes) in the bridge or gateway would be required to address the nodes of the new network. The ease with which a network could be added to an existing module depends also on the accessibility of network cabling. Networks in existing modules, however, could be easily expanded if the wiring space has excess capacity to accommodate the growth.

TABLE I
GROWTH OPTIONS SUMMARY

OPTION	ADVANTAGES	DISADVANTAGES/LIMITATIONS
o Add/Modify Software	o No added HW cost o No added weight, volume, power demands o No HW configuration changes	o Limited by CPU/memory capabilities o SW development cost factors increase significantly as resource limits are approached o Extensive reverification and configuration control required
o Resource Margin	o No HW/SW change until resource limits reached o Probably lowest life cycle cost	o Difficult to predict design point o High initial cost investment o Added initial weight, volume, power o Promotes technology obsolescence
o Technology Insertion	o Improved performance per weight/volume/power o Little impact on existing SW if appropriate standards adopted o Take advantage of technology not available at IOC	o HW cost for space qualification o May be limited by other HW components (i.e., bus bandwidth, etc.)
o Resource Addition	o High potential for enhanced resource capability o Little impact on existing SW except for operating system o Promotes autonomous development, integration, test and operation	o Limited by interconnection bandwidth o HW cost for space qualification o Added weight, power o Network interaction uncertainties
o Network Addition	o Little impact on existing networks and software o Significant capability enhancement o Can be developed and operated autonomously	o May be limited by "bridge" or "gateway" capability to other networks o Added cost, weight, volume, power is significant

2.2.4 DELETED

2.2.5 SYSTEM INTERFACES

2.2.5.1 PAYLOAD/SSDS INTERFACE OPTIONS

1.0 INTRODUCTION

The purpose of this paper is to present options associated with the Payload/SSDS interface.

Certainly some portion, if not all of the interface, would connect to one or more Local Area Networks (LAN). It is assumed that each device will interface with one or more LANs through a standardized Network Interface Unit (NIU). The NIU's may differ in design from the type of devices that are connected to the LAN; NIU's would thus differ for payloads as opposed, to say, printers and other devices. Although payload interfaces will be standardized, there may be different types of interfaces due to variation of payload parameters such as data rate, required ancillary data, on board computer usage, data storage requirements, etc.. This variation suggests the existence of a second device referred to as the Customer Interface Unit (CIU), between the NIU and the payload. This CIU would likely be a part of the SSDS as opposed to being a unique customer design. The options described herein apply to devices such as a CIU or an NIU.

2.0 OPTIONS

Options are divided into technology, design, and programmatic areas. A summary of the options is included in a Table at the end of this section.

2.1 Technology Options

Technology options include intelligent interface devices. Interface Intelligence can range from passivity to a high degree of intelligent decision-making which interacts with other SSDS elements such as the Data Management System (DMS). Two examples of intelligent decision-making options are:

PRECEDING PAGE BLANK NOT FILMED

a. Output control as, for example, data rate based on link bandwidth capacity, LAN overcapacity, etc..

b. Intelligent fault diagnosis.

Output rates from the payload may be controlled at its interface with the SSDS in order to accommodate resource availability and allocations such as link and LAN bandwidths. This may require buffering in the payload and/or control of the payload observation initiation. In either case, this interface communicates with the payload and operates in at least one of two manners: 1) to allow payload output to be ignored if it cannot be accepted by the SSDS or 2) to permit the payload to provide its own storage and playback at an average rate consistent with its duty cycle as the output passes through the interface.

The Goddard Space Flight Center (GSFC) Customer Requirements document stipulates a requirement that the payload interface support the customer by providing some level of fault diagnosis in the event of payload failure. The degree of fault diagnosis support provided by the interface (NIU or CIU) can be implemented through a variety of techniques. For example, on indication of failure the interface could initiate a pre-established set of signals to the payload and transfer return signals directly to the customer, who could interpret these return signals and initiate appropriate corrective action. The interface device could also contain a programmable microprocessor that would initiate signals that were programmed by the customer with the returning signals again passed to the customer or interpreted by the interface device. The means of detecting failures also represent a set of options. For example, the interface device could maintain a continuous monitoring signal directly to the payload, for which the absence of an echo would indicate a failure. Alternatively, the interface device could monitor the payload output and assume a failure in the absence of this output.

2.2 Design Options

2.2.1 Command Management Support and Automated Command Verification

It is reasonable to assume that verification of all commands transferred to a payload would occur at the last step in the SSDS data flow, i.e., at the payload/SSDS interface. There are a variety of options for performing this command verification. Options include sequence counting and recognition of specific commands. Customers who desire privacy may prefer that their command activity not be acknowledged in the clear. This suggests that the interface has programmable verification under customer control. The varied customer mix further suggests that the interface command verification might be programmable and, therefore, capable of being changed for specific customer requirements.

Command verification could be extended to include a variety of options associated with command management. For example, the interface could be designed to provide an inhibit function for restricted and constrained commands. Given this inhibit, the interface could communicate with the DMS (or any other Space Station subsystems) prior to transferring these inhibited commands. This communications could consist of evaluating the current Space Station status in order to determine whether the execution of a constrained command was acceptable. Likewise, restricted commands could be verified for execution within the DMS prior to transfer and execution. This capability would be consistent with the desire to perform command validation at the last point in the command stream.

2.2.2 Data Type Recognition and Routing

The payload output may have different rates depending on the data type. For example, very high data rates, perhaps in excess of 50 Mbps, might be routed directly to a communication and tracking interface as opposed to the LAN. Similarly, payload engineering data might be transferred on a different path prior to transmission. In such cases, the interface could select routing based on recognition of data type. Further, the interface could be designed with sufficient intelligence to select routing based on the network conditions. Failed or temporarily degraded paths might be bypassed.

2.2.3 Packet Construction

The GSFC Customer Requirements state that the payload will construct packets. Packet construction will vary from payload to payload with such factors as the use of secondary data (ancillary data) and the length of the packet. Options associated with the packing of secondary data include 1) allowing the payload to select preferred data from all possible ancillary data or 2) directing preselected ancillary data to the payload by selection on the SSDS side of the interface. In the latter case, this selection could be performed by the CIU or NIU in coordination with the DMS. Another option would be to allow variable length packets for any given payload; with this option, it might become necessary to design the CIU or BIU to contain knowledge of packet length in order to execute the proper protocol with the LAN. This flexibility within that part of the packet used to support LAN communications transfer would allow for LAN technology upgrades or communication system changes without modifying existing or planned payloads to conform to previous interface standards.

2.2.4 Crew Interface

The CIU or NIU could be designed to provide status information on payload conditions to crew members. This status presentation could be standardized for a different set of crew members. Among the options for providing this status information are: physical location (on an interface itself or through query to a remote terminal), packet status operating mode (on, off, standby, failed, etc.), and current activity (queue status, power condition).

2.3 Programmatic Options

2.3.1 Development Responsibility

Any of several NASA organizations could be responsible for interface development. Options include: 1) development by a single NASA organization for all space LAN interfaces including payload interfaces or 2) development by several organizations of the NIU and the possible CIU. As the operation of the payload interface will be closely associated with the transfer of payload data on the ground and to the customer, there are also options associated with development coordination between space LAN interfaces, ground LAN interfaces, and SSDS interfaces to the ground customer. Consequently, there are organizational decisions in the development of these devices which transfer payload data in space or on the ground.

2.3.2 Interfacing Testing Responsibility

The payload interface will require testing in the development stage and prior to operation. Again, options exist for this testing responsibility for all phases of interface development as well as the different operational phases. This responsibility could be centralized within a single organization or distributed among several organizations as discussed in the preceding section.

2.3.3 Payload Certification

Options for certification of a payload for flight readiness include either certification at a central facility (such as a simulation facility), or, given a long distance communications capability, certification at any of the number of remote sites including the customer facilities.

The extent of simulation required to achieve certification also presents a variety of options. Certification may be achieved with a single test or a series of tests that coincide with the payload buildup schedule. The test(s) couia De extremely stringent including, for example, verification of all command sequences including evaluation of potential threats to the SSDS imposed by payload operations.

SUMMARY OF PAYLOAD/SSDS INTERFACE OPTIONS

Option Type	Area	Options Presented
Technology	Intelligent Interface Devices	Control payload output based on SSDS resource availability.
	1) Control Payload Output	Either ignore payload output or buffer it.
	2) Fault Diagnosis	1) Pre-established signal levels 2) Microprocessor control 3) Continuous signal 4) Monitor output level
Design	Command Management/ Command Verification	1) Sequence Counting 2) Command Recognition 3) Customer Controlled Verification 4) Inhibit Constrained and Restricted Commands
	Data Type Recognition and Routing	1) Breadth of Data Types and Rates 2) Network Conditions
	Packet Construction	1) Ancillary Data Usage/Control 2) Packet Length 3) Upgrades Support
	Standardized Crew Interface for Status Information	1) Physical location 2) Packet Status Operating Mode 3) Current Activity
Programmatic	Interface Development	1) Single vs multiple NASA organizations 2) Space LAN/ground LAN/ and customer interface
	Interface Testing	1) Development Stage 2) Operations Stage 3) Centralized/Distributed
	Payload Certification	1) Simulation Control Facility 2) Remote Sites 3) Extent of Certification

2.2.5.2 DELETED

2.2.5.3 Man-Machine Interface (Workstations)

The costs associated with supporting human life in space are very high, and therefore it is vitally important to get the maximum benefit from this human resource. To achieve this goal, the man/machine interface must incorporate several diverse state-of-the-art technologies into a single, coherent, effective system.

The MMI design must consider techniques for workstation automation, reconfigurable display and control systems, caution and warnings, and workstation configuration.

The Space Station RFP (C-4-2.2.5.3) calls for Multipurpose Application Consoles (MPAC) to be a common design functioning as a man/machine interface to the network operating system. The MPAC shall provide command and control, monitoring, operations, and training capabilities. Other required features include visibility into all subsystems, simultaneous viewing of displays, crew override for subsystem operations, annunciation of failures consistent with caution and warning philosophy, and portable capabilities to support both EVA and IVA operations.

2.2.5.3.1 Workstation Automation

2.2.5.3.1.1 Description

The amount of workstation automation provided in the space station must be considered in conjunction with the role of the crew in the operation, maintenance, and use of the station. These considerations will of course be driven by available technology and the basic operational philosophy of the space station. Rapid developments in microprocessor technology and their applications have increased to the point where it is possible to automate the majority of crew workstation functions. There are many questions to be answered when considering whether total system safety and performance is always enhanced by allocating workstation functions to automatic devices rather than human operators. The question today is not whether a function can be automated, but whether it should be, due to various human factors issues.

Crew workstation automation will directly dictate the role of the space station crew. The two options available involving workstation automation and the space station crew are: will the space station crew operate in a highly automated environment and assume the role of system manager or will the environment be less automated, dictating the role of the crew as operators.

2.2.5.3.1.2 Option Characterization

a) OPERATOR PERFORMANCE CHARACTERISTICS	SYSTEM MANAGER PERFORMANCE CHARACTERISTICS
1. Extended periods of valuable crew time are consumed as system operators.	1. Allows for reduced crew workload and additional time for maximizing mission productivity.
2. As operators, crew awareness tends to be focused on workstation.	2. Failure detection must be designed carefully to increase crew awareness.
3. Automatic redundancy provisions are not required.	3. Appropriate fail-safe automatic redundancy provisions must be provided.
4. Manual takeover problems not a major concern.	4. Design of manual takeover of system must not be complex.
5. Crew job satisfaction, and prestige are not threatened.	5. Care must be taken not to impair crew job satisfaction, self-con-concept and prestige.
6. Automation induced failures are of little concern.	6. Complex verification procedures must be incorporated to eliminate automation induced failures.
7. Crew error rate higher, resulting in lower safety factor.	7. Crew errors are reduced, thereby enhancing system safety.
8. Extensive memorization and use of cue cards required.	8. Reduces crews needed for memorization and use of cue cards.

a) OPERATOR PERFORMANCE CHARACTERISTICS	SYSTEM MANAGER PERFORMANCE CHARACTERISTICS
9. Lower overall performance levels.	9. Higher overall performance levels.
10. Maintainability of manual skill levels.	10. Possible loss or degradation of manual skill levels.
11. Integrity of input data discernable by operator.	11. Integrity of input data must be high.
12. Relatively straight forward software and hardware development.	12. Extremely complex software and hardware development.

b) OPERATOR PROGRAMMATIC CHARACTERISTICS

1. Cost will be moderate for operator concept and manual workstation.

2. Manual workstations have been developed and used; development schedules will reflect same.

3. Technology is available and is in a mature state.

4. System evolution will consist mainly of refining operating procedures.

SYSTEM MANAGER PROGRAMMATIC CHARACTERISTICS

1. High cost will be associated with the system manager concept and workstation automation.

2. Automated workstations will be a relatively new technology and schedule constraints must reflect same.

3. Technology is available, although more applications work on automated systems needs to be done.

4. System will evolve from technology advances such as artificial intelligence.

c) RISK ASSESSMENT

1. Option #1 - Operator

This is a level 7 technology readiness level; Engineering model tested in Space. Skylab and space shuttle are operator type systems.

2. Option #2 - System Manager

This is a level 3 technology readiness level; Conceptual design tested analytically or experimentally. Automated commercial airline cockpit workstations have been developed and are currently used experimentally.

2.2.5.3.1.3 Projected Capabilities

a) Option #1 - Operator

1987 - Operator technology is mature. Skylab, space shuttle, and commercial airline cockpits are operator type systems.

1995 - Operator technology will be mature with respect to space station operation.

2000 - Strides in advanced operator technology with respect to the space station will be made. Operating procedures will be optimized and refined to the smallest detail.

b) Option #2 - System Manager

1987 - System manager technology will be formulated with respect to the space station. Automated procedures will be incorporated at the basic technology level.

1995 - System manager technology will be advanced through growth and evolution by more automation being incorporated into the crew workstations. Expert systems will begin to emerge in system control.

2000 - Automation technology will have advanced to a star wars type sophistication. Technologies such as voice control and artificial intelligence will be incorporated as an integral part of system operation.

c) Key Drivers

The key driver to the operator option will be experience gained on the space station and experiments performed on ground in crew workstation mockups.

Initially, key drivers to the system manager technology may well be the area of commercial cockpit technology. Many of the areas in commercial airlines cockpit operation will be applicable to space station control. In order to reach the sophistication desired for future space station operation, technology areas such as voice recognition and artificial intelligence must be advanced greatly.

2.2.5.3.1.4 References

1. Human Factors of Flight-Deck Automation - NASA/Industry Workshop, Deborah A. Boehm-Davis, Renwick E. Curry, Earl L. Wiener and R. Leon Harrison, January 1981.

2. Karl Chase, Staff Engineer, Sperry Flight Systems

2.2.5.3.2 Multifunction Display

2.2.5.3.2.1 Description

Man wants to see and hear as much as he can of the things that concern him, and he becomes anxious when deprived of adequate information about what is going on about him. Consequently, he should be provided all information relevant to his job. All information need not be displayed all the time, but when desired, he should be able to call up a subsystem situation display which contains all system parameters and trends. For the space station and its immense amount of sophisticated data, the crew members will require an intelligent method to integrate and absorb information. The multifunction display appears to be the prime consideration for solving this problem.

There are two basic kinds of display: symbolic and pictorial. In symbolic displays, the information presented has no pictorial resemblence to the conditions represented. Pictorial displays do have a pictorial, geometrical or schematic resemblence to the things they represent. The question arises whether including color in a multifunction visual display will help maximize workload reduction, performance and safety. The two options to be discussed here are the use of monochrome or full color multifunction displays for the space station crew workstation.

2.2.5.3.2.2 Option characterization

a) MONOCHROME PERFORMANCE CHARACTERISTICS	COLOR PERFORMANCE CHARACTERISTICS
1. Information enhancement limited to format manipulation only.	1. Color offers potential benefits by providing an additional dimension for color encoding high density information.
2. Clutter can result from improper format design.	2. Excessive use of color can interfere with interpretation of displayed information.
3. Human factors well established for monochrome displays.	3. Need to develop a color human factors data base from human performance testing.
4. Monochrome performance degradation slight in high ambient lighting.	4. Color performance degrades in high ambient lighting.
5. Regulations well established for monochrome displays.	5. Lack of regulations exist for color displays.
6. Monochrome CRTs have higher efficiency than color CRTs.	6. Color tubes have lower efficiency than monochrome CRTs.
7. Monochrome CRTs have high resolution capability.	7. Resolution on color tubes limited both geometrically and electronically.
8. Monochrome CRTs are highly reliable.	8. Color CRTs are less reliable then monochrome CRTs.
9. Flat panel technology has developed workable monochrome displays.	9. Flat panel technology does not have a useable full color display as yet.

b)

MONOCHROME CRT PROGRAMMATIC CHARACTERISTICS	COLOR CRT PROGRAMMATIC CHARACTERISTICS
1. Cost is low for monochrome displays.	1. High cost is associated with quality color displays.
2. Monochrome displays require only format development and design. Scheduling will not be constrained.	2. Color displays require format development and design plus color coded information and human factors analysis. Scheduling will be moderately extended.
3. Technology is available and in a mature state. Flat panel monochrome displays are rapidly becoming available.	3. Technology is mature for CRTs but may not be available in flat panel for the initial operating configuration of the space station.

c) RISK ASSESSMENT

1. Option #1 - Monochrome

This is a level 6 technology readiness level; Prototype/Engineering Model tested in relevant environment. The space shuttle has a monochrome (IBM MCDS) multifunction display system in use.

2. Option #2 - Color

This is a level 4 technology readiness level; Critical function/ characteristic demonstration. Commercial airlines currently use full color CRTs in their flight management systems.

2.2.5.3.2.3 Projected Capabilities

a) Option #1 - Monochrome

1987 - Monochrome technology is mature. Space shuttle uses monochrome CRT. Flat panel monochrome display technology has progressed to a point where incorporation into the space station is feasible.

1995 - Monochrome flat panel technology has progressed to the point where they can be definitely incorporated into the space station. Monochrome formats for the space station will be refined to their optimum.

2000 - No significant change from 1995 has taken place except more efficient and "better" monochrome flat panel technology will have been developed.

b) Option #2 - Color

1987 - Flat Panel Color CRT technology is not mature enough to use on the initial operating configuration of the space station. Human factors with respect to color technology is becoming mature.

1995 - Full color flat panel technology is mature enough to use on the space station. Human factors with respect to color has been refined and in use.

2000 - The use of color in the space station multifunction displays will be highly refined. The human factors aspect will be well defined from experience gained on the space station and simulator studies.

c) Key Drivers

The key drivers to monochrome multifunction displays will be the development of flat panel technology.

The key drivers to color multifunction displays will be the development of flat panel technology and the human factors data base for color use.

2.2.5.3.2.4 References

1. "Color CRT Display for the Cockpit", Harry L. Waruszewski

2. Karl Chase, Staff Engineer, Sperry Flight Systems.

2.2.5.3.3 Caution/Warning Techniques

2.2.5.3.3.1 Description

The space station advisory, caution, and warning system design must incorporate the basic operational philosophy of the space station as well as the conceptualized role of its crew members. Processing will involve determining other-than-normal conditions of the entire core space station systems.

Alerts must be prioritized in two forms. First the alerts must be grouped into the critical categories of warning, caution, and advisory alerts. Secondly, the alerts must be ranked in order of importance so that when several alerts are activated simultaneously the more urgent alerts suppress the less urgent alerts. In order to prepare the crew for a problem before it reaches a critical level, automatic trend analysis must also be performed by the advisory, caution, and warning system.

In order to avoid crew member saturation and/or confusion, the overabundance of caution, warning and advisory alerts must be avoided. It is suggested that both visual and aural message outputs be incorporated into the system.

Two options exist when considering an advisory, caution, and warning system design. The system may be a distributed type system or a single integrated system.

2.2.5.3.3.2 Option Characterization

a) DISTRIBUTED ADVISORY, CAUTION, AND WARNING SYSTEM PERFORMANCE CHARACTERISTICS	INTEGRATED ADVISORY, CAUTION, AND WARNING SYSTEM PERFORMANCE CHARACTERISTICS
1. Has tendency to proliferate workstation with alerts.	1. Alerts are more easily consolidated.
2. Difficult task to correlate alert-type applications and significance.	2. Central processor software can easily correlate alert-types and their significance.
3. Almost impossible to categorize and prioritize alerts.	3. Central processor software improves ability to categorize and prioritize alerts.
4. Inhibiting alerts not directly controllable.	4. Non-critical alerts may be inhibited during a period of high workload.
5. Relatively straight-forward hardware development.	5. More complex hardware development.
6. Relatively straight-forward software development and integration.	6. Extremely complex software development and integration.
7. Lower overall crew member performance level due to proliferation and non-categorization of alerts.	7. Higher overall crew member performance level.
8. Correlation of alert and system checklists not straight-forward.	8. Facilitates correlation of alert and system checklists needed for checkout.
9. Alerts can be added on a modular basis.	9. Integrated system hardware/software must be modified to add, delete, or alter alerts.

b)

DISTRIBUTED ADVISORY, CAUTION, AND WARNING SYSTEM PROGRAMMATIC CHARACTERISTICS	INTEGRATED ADVISORY, CAUTION, AND WARNING SYSTEM PROGRAMMATIC CHARACTERISTICS
1. Cost will be moderately high for distributed system due to hardware and processing redundancy.	1. Although hardware may be reduced for the integrated system, software and systems development will create a high cost.
2. Distributed advisory, caution, and warning systems are not complex and should not create schedule constraints.	2. An integrated advisory, caution, and warning system will require large amounts of system and software development time thereby increasing schedule development time.
3. Technology is available and in a mature state.	3. Technology is available, although much applications, systems integration, and software work will need to be done.
4. System evolution will consist of modular additions and alterations.	4. System evolution will consist of expanding and revising integrated system.

c) RISK ASSESSMENT

1. Option #1 - Distributed Advisory, Caution, and Warning Systems.

   This is a level 6 technology readiness level; Prototype/Engineering Model Test in relevant environment. Skylab and space shuttle have distributed caution and warning systems.

2. Option #2 - Integrated Advisory, Caution, and Warning Systems.

   This is a level 2 technology readiness level; conceptual design formulated. Basic designs have been formulated for the commercial airline industry although they have not been implemented and tested in cockpit mockups as yet.

2.2.5.3.3.3 Projected Capabilities

a. Option #1 – Distributed advisory, caution, and warning system.

1987 – Distributed advisory, caution, and warning systems technology is mature. Skylab and Space Shuttle are the basis for space station design decisions.

1995 – Distributed advisory, caution, and warning system will have matured with respect to the space station operational philosophy and crew members role. System procedures and anamolies have been satisfactorily been resolved.

2000 – Additional modular advisory, caution, and warning systems will be added to enhance growth and evolution. Existing systems will be refined based on space station and ground simulator experience.

b. Option #2 – Integrated advisory, caution, and warning system.

1987 – Integrated advisory, caution, and warning system technology will be formulated with respect to the space station. Basic technology will be ready to incorporate into the space station fulfilling its basic requirements.

1995 – Integrated advisory, caution, and warning technology will have advanced significantly through experience gained on the space station and on ground simulation.

2000 – Technology will be advanced and refined to the smallest detail. The system will have been modified and expanded considerably.

c) Key Drivers

Initially, the key drivers to the integrated advisory, caution, and warning systems will be from the area of future commercial cockpit experimentation. Experience will also be gained onboard the space station and from future crew workstation mockups. The main area of importance will be in the area of caution and warning systems software and applications.

The key drivers to the distributed advisory, caution, and warning system will be the technology formulated by the commercial airline industry and results obtained in the Space Shuttle and Skylab. Voice synthesis will need to be developed for these systems.

2.2.5.3.3.4 References

1. Design Criteria for Aircraft Warning, Caution, and Advisory Alerting Systems, J. E. Vietengruber, 1977.

2. Karl Chase, Staff Engineer, Sperry Flight Systems

2.2.5.3.4 Workstation Configuration

2.2.5.3.4.1 Description

Although conventional cockpit technology has served us well in the past, the advent of the space station has increased the complexity and sophistication of the information that will be needed to properly control, operate and maintain this vehicle. New approaches are needed to achieve higher levels of safety, efficiency, and performance required for earth orbital control and long term sustentation. Dramatic technological improvements are creating the potential for vast improvements in spacecraft crew workstations.

Because of the need for new and more sophisticated information in the crew workstation, the space station crew members will require an intelligent method to integrate and absorb information. The efficient use of central computation and integrated multi-mode electronic displays is appropriate.

The first decision concerning the space station workstations will be that of configuration. Two options exist for the space station crew workstations, these are: a centralized crew workstation, or distributed crew workstations through the space station.

2.2.5.3.4.2 Option Characterization

a) CENTRALIZED CREW WORKSTATION PERFORMANCE CHARACTERISTICS	DISTRIBUTED CREW WORKSTATION PERFORMANCE CHARACTERISTICS
1. Centralized work crew members to populate one module to use work station.	1. Distributed work crew members to perform duties at different work stations in different modules.
2. All display functions must be incorporated into a centralized display set.	2. Display functions may be split among distributed crew station displays.
3. All cabling will interconnect to centralized crew workstation.	3. Extra cables and interconnects must be used to distribute information to the distributed crew work.
4. The weight of a centralized crew workstation will be less than the sum total weight of the distributed systems.	4. The sum weight total of the distributed systems will exceed that of one centralized system.
5. Centralized workstation will have a high data rate.	5. Distributed workstation will reduce data rate.
6. Computational power must be higher for centralized workstation.	6. Computational power is lower for distributed workstation.
7. Development of centralized workstation complex.	7. Development of distributed workstation not as complex as centralized workstation.
8. Growth and evolution requires modification of entire centralized workstation.	8. Growth and evaluation may be done piecemeal through additions of more distributed workstation.
9. Controlling software extremely complex.	9. Controlling software moderately complex.

b) CENTRALIZED CREW WORKSTATION PROGRAMMATIC CHARACTERISTICS

1. Cost for centralized workstation should moderately exceed that of a distributed crew workstation. Mainly due to complex hardware and software development cost.

2. Schedule constraints for a centralized workstation will moderately exceed that of a distributed crew workstation. This is again due to the complex hardware, software, and integration.

3. Technology for a centralized crew station is currently being refined. Mainly this is being done for commercial airline cockpits.

4. System evolution will consist of of modifications, deletions, and additions to centralized processor.

DISTRIBUTED CREW WORKSTATION PROGRAMMATIC CHARACTERISTICS

1. Cost for a distributed workstation should not exceed that of a centralized workstation.

2. Development scheduling for a distributed workstation should not exceed that of a centralized workstation. But distributed information processing department, and development of different configurations is needed for each distributed station.

3. Technology for distributed crew work station has not been an area of high priority.

4. System evolution and growth will be enhanced by modular additions of work stations as an evolutionary process.

c) RISK ASSESSMENT

1. Option #1 - Centralized Crew Workstations

This is a level 4 technology readiness level; critical function/ characteristic demonstration. Centralized cockpits for commercial airlines have been developed and demonstrated. Much of the work done in human factors and ergonomics will be applicable to the space station crew workstation.

2. Option #2 - Distributed Crew Workstations

This is a level 1 technology readiness level; basic principles observed and reported. Low priority has been given by industry to the distributed workstation concept. The space station will be the first area of applicability.

2.2.5.3.4.3 Projected Capabilities

a) Option #1 - Centralized Workstation

1987 - The centralized crew workstation concept will be conceptually designed and analytically tested. Much human factor data is gathered from commercial airline cockpit design and experimentation has begun in those area where commercial cockpit design is not applicable.

1995 - A centralized crew workstation is operational in the space station. The centralized workstation is continually being experimentally tested onboard the space station and in on ground simulators.

2000 - Centralized crew workstation human factors data base is well refined, and incorporates sophisticated automation.

b) Option #2 - Distributed Workstation

1987 - The distributed workstation concept has begun its conceptual design. Much work is yet to be done in the areas of information networking, human factors and task partitioning.

1995 - A distributed crew workstation is operational in the space station. Human factors data base is being expanded through on board experiments and ground simulation.

2000 - Distributed crew workstation data base is well established. New distributed workstations have been installed in module additions.

c) Key Drivers

Initially, the key drivers for the centralized crew station will be the human factors data base and technology gained from the future commercial airline cockpit mockups and simulations. In future years, on board space station experience and ground simulator experience will provide the majority of data needed to drive the centralized crew station concepts. Artificial intelligence, voice recognition systems, and flat panel displays are the main technology drivers.

The key drivers to the distributed crew station concept will be the same as described for the centralized crew station since the distributed system will be subsets of centralized concept. An additional driving technology for distributed crew stations will be information networking.

2.2.5.4 DELETED

PRECEDING PAGE BLANK NOT FILMED

SPACE STATION DATA SYSTEM (SSDS)
SECURITY/PRIVACY OPTIONS

2.3 DATA SYSTEM SECURITY/PRIVACY (The procedures and controls needed to protect and to limit access to the data system).

OVERVIEW COMMENTS: The Space Station Data System (SSDS) will perform many vital functions and support many diverse users. Key requirements discussed in this paper are:

1) A failsafe and secure SSDS to assure the health and safety of the Space Station (SS) crew and of the Station itself. Assured communications between the Station and the ground and access to required data and apparatus control via the Data System will be needed, especially in times of emergency.

2) The diverse security/privacy requirements for customer data and commanding which will be needed to attract a broad spectrum of Space Station users.

The primary requirement discussed in the NASA Goddard Space Flight Center (GSFC) Customer Requirements for Standard Services from the Space Station Information System (SSIS) document (Reference 1) is to implement data privacy. Data privacy implies control of access to customer data (and related operations), but no complete security guarantee. Customers desiring data and information security must assume this responsibility, including the responsibility for data or command encrypting and decrypting. The SSDS will, however, support and be compatible with customer encryption and decryption equipment.''Three broad categories of data system security/privacy are discussed -

The first deals with SSDS access authorization (or denial). The second deals with data and user authentication, i.e., assurance that the data sender is who he says he is and that the data has not been modified before receipt by the user. The third deals with possible denial of service, i.e., assuring reliable and available data system services when required.

Most data systems deal primarily with the first of these categories. The last category, service availability and integrity, is typically the least implemented aspect of data system security. This paper focuses on the first category (data and data system access restrictions and authorizations), but authentication concerns and the assurance (or denial) of data system services are also addressed.

Section 2.3.3 (operating system security) and Section 2.3.5 (data encryption techniques) present specific information on established techniques and systems. Since many data system

PRECEDING PAGE BLANK NOT FILMED

security/privacy options are dependent on policy or programmatic decisions, other sections are more discursive. These sections allude to the types of security/privacy decisions which need to be made, or present optional policies which need to be determined before specific SSDS design decisions can be made. In many areas it is premature to evaluate specific system design options until higher level policy decisions have been made.

Within this context, the most relevant options are presented for assuring 1) data privacy, 2) the necessary SSDS security for crew and Station safety, and 3) data system integrity.

References are listed at the end of the paper.

2.3.1 DATA PRIVACY APPLICABILITY POLICIES

2.3.1.1 Data Types

2.3.1.1.1 Description. Different categories of data (and associated users) require differing levels or focuses of privacy/security. Among the categories are: 1) scientific/technical data; 2) commercial proprietary data; 3) ancillary/engineering data; 4) foreign user data; 5) scheduling data; 6) command and control data; and 7) potential (post IOC?) national security data. National security requirements will not be discussed, since the military has not yet expressed formal interest in use of the Space Station. It is important to design a system compatible with and upgradeable to possible future military requirements, however.

1) The scientific/technical Space Station user often will have the least need for high levels of security/privacy.

First, much scientific data is not readily useable by others. Often significant algorithmic data processing is required, wherein many particulars of the experimental apparatus and conditions, as well as detailed knowledge of the science, are needed. This implies that few other scientists could even use the data. Furthermore, the computational expense required to digest and meaningfully extract information from the data is often significant and not readily available.

Second, some Space Station experiments involve data rates of hundreds of megabits/second. The data system will be sufficiently taxed to simply capture, preprocess, minimally archive, and distribute this amount of data, without implementing encryption and decryption as well.

Finally, many scientists are unfamiliar with implementing encryption protection. Since their data is generated for eventual distribution within the public domain, there is little desire or precedent for data encryption. Data privacy and supporting policies to permit the Principal Investigator to analyze results first are necessary, however.

2) Commercial users will, however, require sufficient privacy/security to assure themselves that no potential competitor can obtain their data. The commercial user will impose high demands on the SSDS with respect to privacy/security. The SSDS design must meet their needs if the Space Station program is to attract them.

The commercial user can provide ultimate data protection by encryption (see Section 2.3.5). A primary design requirement (Reference 1, 5.4.4) of the SSDS is to provide standard interface(s) to customer provided encryption equipment. Many commercial customers may not want to implement their own encryption, however, if they have adequate assurances of a reasonable level of privacy protection within the SSDS.

3) Data privacy protection of the supporting Space Station ancillary/engineering data may be more troublesome. Commercial customers will often want data privacy with respect to their use of SSDS core and standard user resources (power, computational resources, vacuum facilities, heating sources, etc.) since potentially useful information about their investigations might be inferred from this type of data. Information may even be inferred from the time sequence with which the various Space Station resources were utilized. Other Space Station users, however, may legitimately need to know the environmental conditions within the Station at the time particular aspects of their experiments are being or were performed. Furthermore, there will be a tendency to want to make standard resource utilization records generally available to all Space Station users. Optional procedures to address the problem of ancillary/engineering data dissemination are addressed in the next subsection.

4) A potential area of concern regarding data privacy/security are the requirements of foreign users. The Space Station program has made a priority of soliciting foreign involvement and commitment. Firm commitments with foreign governments are being negotiated. The privacy requirements of foreign users are not addressed in this paper. It has been assumed that they have no more stringent requirements than that imposed by the U.S. commercial customer desiring protection of his proprietary data and his accompanying operations.

5) The protection of scheduling data is important both for customer privacy and for Space Station system safety. The issues here are closely related to those for commanding, since schedule

data will eventually be translated into command data for system resources, etc.

6) Command and control validation and authentication present different areas of concern. The desire to implement transparency of customer commanding can seriously conflict with the need for Station safety. This issue is addressed in Section 2.3.5.

2.3.1.1.2 Option Characterization. Options to assure adequate SSDS operational security and data privacy are the focus of the rest of this paper. One option discussed here is privacy related to ancillary/engineering data distribution.

One approach is to require selected subsets of the ancillary/engineering data to be restricted to all but authorized users. This approach may prove unacceptable because of the difficulties of determining whom should be authorized and for what data subsets; it is also difficult to assure that authorized users will secure the data they have so obtained. Decisions as to which ancillary data might be restricted (not generally needed by other Space Station users) are difficult.

An alternative approach is to inform commercial and other users sufficiently in advance that ancillary/engineering data sets will not be restricted. They can then choose to design their apparatus to conceal the usage (or timeline of usage) of particularly sensitive resources. For example, a commercial user requiring large surges of power periodically can disguise his experiment to request moderate power usage over a longer period of time and store this energy (in capacitor banks, etc.) to produce the needed power surges. The power request timeline can be concealed in a similar manner. An accurate power request log could thus be maintained by the SSDS and disseminated freely to all users. Similarly, TDRSS downlink data can be filled to always be at a 300 MBPS/sec rate. Not only might this help conceal actual data generation characteristics,but may reduce system reconfiguration procedures required for changing downlink rates.

Personnel within the Space Station program, however, may still need to know what the commercial customer is doing if, for example, his power surges generate sufficient electromagnetic disturbances to affect other users. Similarly, if stray electromagnetic field measurements are provided as part of the generally available ancillary data, then a customer has the option to provide shielding to hide his operational use of power. If a policy on what ancillary data (if not all) will be generally available is provided sufficiently in advance, the commercial customer can usually adapt his experimental apparatus or techniques to provide a high level of operational privacy. This option may be more implementable as a general policy than an attempt to restrict access to subsets of ancillary data.

2.3.1.2 Data Rate Considerations.

2.3.1.2.1 Description. Space Station system users will generate a wide range of data rates. High data rate users (above 10 Mbps rate) will have different needs and constraints than moderate or low data rate users. This suggests advantages for two or more logically separate downlinks.

2.3.1.2.2 Option Characterization. The amount of data generated may affect the types of privacy/security options available. User data rates of 100Mbps, for example, may not be easily encrypted without significant data compression.

One policy alternative is for the SSDS to provide an optional standard encryption service. Some users may not want to provide their own equipment, but would utilize its availability (at modest cost), and the higher level of data access privacy which encryption would support. The user can provide higher protection himself since the security of the SSDS encryption/decryption key(s) would presumably be less than that which an individual user could implement. An optional SSDS encryption service might be highly desirable for users generating minimal amounts of data, and for whom security requirements are moderate. Encryption techniques and options are presented in Section 2.3.5.

2.3.1.3 Network Location.

2.3.1.3.1 Description. There will be different data and command privacy needs onboard versus on the ground, and among different nodes in the SSDS network. Alternative policies of coordinating the overall privacy/security requirements of the network versus at each node are related to the decision whether to institute an end-to-end security policy or have individual implementations at different nodes within the network.

With encryption an end-to-end security/privacy implementation is usually adequate since any intercepted data is meaningless at an intermediate point (if it is encrypted at the source and decrypted at the destination). Intermediate checks on the maintenance of data integrity (no unauthorized modification, etc.) need to be implemented at intermediate nodes even in this case, however.

2.3.1.3.2 Option Characterization. Key options onboard, on the ground, and among network nodes (or end-to-end) are delineated below:

Onboard: Onboard customers or crew will have been preselected and screened carefully with respect to integrity and perceptiveness to the need for data system privacy and security. There is little likelihood of stowaways or infiltrators among this group. There is always a possibility of one customer attempting to listen to another customer's private conversations, however.

The primary threats to onboard system security and privacy will likely be within the command and communications system and/or due to careless errors or subsystem failures onboard. The options for dealing with careless errors or component failures involve designing a failproof system with high system integrity and redundancy. The options addressing these design issues are discussed in Sections 2.3.3, 2.3.4, and 2.3.6.. The options for safeguarding the communications links are discussed in Section 2.3.1.4 and 2.3.6.3.

Audio and video interception is another area of privacy/security concern. Audio and video data will likely be digitized, compressed, and multiplexed with other SSDS data. Techniques and options to deal with the protection are not addressed in this paper.

Ground: In addition to the above onboard security/privacy issues, threats from a wider spectrum of people - commercial competitors, saboteurs, foreign agents, hackers, etc. arise on the ground. A wider spectrum of users may also concurrently address the SSDS for data or services from ground nodes. A higher level of system physical, operational, and privacy protection therefore may need to be implemented within the ground support system.

Erroneous and dangerous commands generated by saboteurs or pranksters are much more likely to be initiated within the ground network. Likewise, commercial competitors are more likely to attempt to gain private information by attempting to foil the ground data network or the communications links. Another area of concern is the prevention of data duplication (hardcopy, tapes, etc.) on the ground. Options addressing some of these issues are presented in the sections listed above for onboard, plus Sections 2.3.2.3, and 2.3.7.

Node versus End-to-End: A successful end-to-end security policy requires system users to understand and have confidence in the policy and its implementation. A user at an intermediate node may not trust the security of procedures for data entering or leaving his node. An optional implementation of system security can be to assign this responsibility to personnel or subsystems at each node. The advantage of this is a feeling of greater data handling security due to local control. Whether enhanced security can be obtained with local control needs to be assessed within the SSDS environment. End-to-end security/privacy implementation has the advantage of relieving intermediate users of this responsibility, except for the particulars of how they handle and safeguard data at their respective nodes.

Assurance that no interruption or data eavesdropping has occurred during internode communications still remains a problem, unless data is encrypted. Users concerned to this degree with overall system security will need to provide their own encryption and decryption equipment or processors. A high level of data system security will not be guaranteed by NASA; data privacy and data integrity will be preserved to the extent feasible and practicable within the SSDS.

2.3.1.4 Communication Link Privacy Options

2.3.1.4.1 Description. Data and commands can be intercepted or interfered with during transmission between different network nodes. This topic encompasses the three basic security areas - unauthorized access to the data (copying, eavesdropping, etc.), unauthorized data modification, and potential denial of service (interference with the communications process).

Discussion of these problem areas and optional approaches to dealing with them are presented below:

2.3.1.4.2 Option Characterization.

Unauthorized Access: It is usually not feasible to prevent interception of the airborne electromagnetic transmissions. The data content can be concealed via encryption, however.

For hardware or data bus transmissions, electromagnetic shielding and/or monitoring of physical taps into the media can minimize privacy breaches. Onboard data subsystems, in particular, may want to incorporate this protection. Assuming TDRSS interception can not be prevented, such shielding may not be as critical on the ground, except subsequent to data decryption at a ground node and transmission to other nodes.

Potential onboard tapping into data buses or hardwire connections can be screened during prelaunch checkouts and by selectivity and training of the onboard crew and mission specialists. The use of optical data buses can also eliminate the potential information loss due to physical tapping into the transmission medium.

Further investigation is needed to evaluate the relevance to the Space Station program of the recent (September, 1984) Presidential Directive Number 145 and its implications regarding data encryption for protection of Space Station program information. One SSDS option may be to preferentially encrypt only low or moderate data rate downlink data. The encryption might then be performed either as a standard service (at optional cost) or by the customer.

Unauthorized Modification: Data tampering or destruction may be particularly serious if commands were modified and/or inadvertently executed, with resultant damage to a payload or jeopardy to crew safety. Similarly if a valid emergency command procedure were destroyed.

Encryption should minimize data modification possibilities; encrypted data can be modified, although the result will normally be garbage. The recipient, after receiving and evaluating the distorted message, can request retransmission. Procedures and protocols to assure adequate and authenticated retransmission therefore need to be formulated and integrated into the SSDS design.

Denial of Service: The disruption of data transmission links can occur periodically due to natural phenomena (intense electrical storms, apparatus failures, etc.)

Four options dealing with interruption of communication link availability or adequacy are:

1) emergency (or standard) alternative communication paths;
2) emergency substitute procedures to replace the service temporarily disrupted;
3) sufficient onboard data buffering to cover the period of disrupted service; or
4) acceptance of a temporary loss of data or service.

The key issue is to evaluate which crucial services or data are likely to be impacted by a failure of TDRSS or other communications links. Risk assessment of both the likelihood of disruption of service and the likely service downtime need to be performed. Reference 3 (pg. 1.2, paragraphs 1.3, 1.5) states that a detailed risk assessment will be performed for use by the Phase B Space Station contractors. A recent contract has been awarded by Kennedy Space Flight Center to perfrom an indepth risk and threat assessment for the Space Station program.

An important option is the implementation of alternative communications routes and paths for all crucial SSDS services. Customers likewise need to develop contingency plans for temporary loss of communications with their payloads. Backup stored onboard command sequences, perhaps leading to automatic shutdown of their payload, are needed in the event that "realtime" commanding from the ground becomes significantly disrupted. Techniques for detecting communications disruption also need to be refined.

2.3.1.5 Summary of Options/Issues

Section	Option/Issue Focus	Scope of Options/Issues Addressed and to be Addressed
2.3 Overview	Categories of Security/ Privacy	Data and Data System Access Denial Data and Sender Authentication Denial of Service
2.3.1		
2.3.1.1 least to	Applicability of Privacy to Data Types	Scientific: probably stringent requirements Commercial: probably most stringent requirements. Data Encryption is customer option. Encryption may be worthwhile optional SSDS standard service. Ancillary/Engineering: Options: 1) restrict access of selected subsets to certain users; 2) no access restrictions inform users sufficiently in advance that they can mask usage of critical resources. Foreign: Not yet specified. Probably no more stringent than commercial. Military: Not addressed. Design SSDS to be compatible upgrades to military-level security.
2.3.1.2	Data Rate Privacy Considerations	High (above 10 MBps) Moderate and Low (below 10 MBps) High: Encryption may not be feasible. Password protection applicable. Low/Moderate: Encryption feasible May not be desired if SSDS sufficiently private. Optional SSDS encryption service may be useful.

2.3.1.3	Network Location Privacy Considerations	Onboard Versus Ground
		End-to-End Security/Privacy Policy vs. Intermediate Node Implementation
		Onboard: probably fewer security threats due to selectivity of crew
		Ground: Broader spectrum of possible security violators
		End-to-End: simplifies security/ privacy control
		Intermediate Nodes: May enhance feeling of privacy implementation due to local control.
		Encryption would minimize communications eavesdropping concerns.
2.3.1.4	Communications Link Privacy Options	Airborne transmissions: 1)conceal data via encrytion 2) specialized receivers
		Hardwire transmissions: 1) EM shielding 2) onboard crew selectivity
data		Presidential mandate regarding encryption of downlink needs study.
		Denial of Service: 1) emergency communication paths; 2) emergency procedures; 3) data buffering; and 4) temporary loss of service
		Risk assessments need to be performed. Backup stored command sequences an option.

2.3.2 Data Systems Access Controls

This section discusses four categories of options to limit access to the Space Station Data System. The first option is the use of passwords. The second category are options to prevent modification of system data. The third set of options are checks on 1) the physical device; 2) the facility from which commands are issued; and 3) the personnel themselves. The final option category deals with the selection and monitoring of data system personnel.

2.3.2.1 Password Options

2.3.2.1.1 Description. Many types of passwords or codes can be used to limit computer system access. Their purpose is to permit only authorized persons access to the computer system, its data, or its services. The computer system maintains an internal, protected directory of these passwords. The system checks the passwords before authorizing access to various services or data.

2.3.2.1.2 Option Characterization. Different types of passwords can be implemented. For low security systems, passwords are usually simple and easy to remember, such as the user's name, date of birth etc.. Although easy to remember, they are also easy to guess by someone attempting to break the system's security. More complicated passwords (numerical sequences, sentences, etc.) are harder to guess, but are often written down, thereby reducing their protection. A query system can also be implemented, whereby a user has to answer particular questions about himself. Password protection is a key concern when it represents the primary security mechanism.

One option is the breadth and scope of password applicability. In addition to requiring passwords for logging onto the system and accessing services, passwords can subsequently restrict access to particular data files. An additional control is to limit the number of system logon attempts. People randomly guessing passwords will be automatically rejected after a set number of logon attempts. A security officer should be notified in this case.

Another option is whether the passwords at the onboard and ground SSDS nodes will be the same or different. Greater password commonality results in easier use, but reduced security. One can implement common passwords for logon but distinct passwords for data file access.

The regularity of password updates is an important option. As SSDS personnel and security requirements change, this regularity may be affected. One option is to allow each user to change his passwords in addition to periodic system—wide changes.

An additional level of password protection can be achieved by encryption of the passwords themselves. This might permit the user to control his own passwords. The use of data encryption to enhance system security is discussed further in Section 2.3.5.

2.3.2.2 Data

2.3.2.2.1 Description. In addition to general system access controls, the SSDS needs to limit access to the data itself. Most computer systems require some access limitation in reading and/or writing data. There are usually different security or access levels to different data sets. In addition to user data and programs, the computer system utility functions need to be protected, i.e., a person invoking an editor program is not able to modify its contents. Protections against a "Trojan Horse," where an editor-type program actually performs secret data modifications or access, also need to be implemented.

The Bell-LaPadula Security Model, discussed in Section 2.3.3, allows read access only to objects at the same or lower security levels. One cannot read a file classified at a higher security level. The so called star property (*property) of this model restricts write accessibility to lower security levels. This prevents a higher security user from transferring unauthorized information to a lower level. The operating system normally verifies the required access controls and data transfer authorizations.

2.3.2.2.2 Option Categorization. Among the options are which users require only write access protection versus those who require read access restrictions as well. The password protection techniques may depend on specific user requirements, and on the amount, type, and location of user data. The ease of SSDS access needs to be traded off against security requirements.

2.3.2.3 Device/Personnel Controls

2.3.2.3.1 Device Type Check Upon Logon

2.3.2.3.1.1 Description. Besides passwords, the system can check that a "logon" is being entered on proper equipment. This can be valuable where many network devices, some at remote locations, have accessibility to the data system. The system can check that the particular hardware characteristics of the logon device meet certain specifications.

2.3.2.3.2 Device/Physical Facility Checks

2.3.2.3.2.1 Description. In addition to the physical device, the system can check the physical facility from which the logon attempt has been made. A signal or message can be sent back (callback) to the device at the particular physical facility to verify its attempted access. Most secure physical facilities also require personal magnetic strip access cards, in conjunction with the requirement to key in a personal identification number (PIN).

Security procedures can also be implemented to prevent user departure from a restricted location in the physical facility until all devices have been signed off. Restrictions on which time periods during the day access is authorized, can also be imposed.

2.3.2.3.3 Personnel Authentication Beyond Passwords

2.3.2.3.3.1 Description. In addition to physical device and facility authentication, the system may check particular personnel characteristics.

2.3.2.3.3.2 Option Categorization. Four means by which personnel may be differentiated and authenticated are fingerprint (or handprint) discrimination, handwriting, voice, and video analysis.

Fingermatrix, Inc. (White Plains, NY) markets a biometric handprint device. Reliability is good, although hand cuts have caused occasional problems. Handwriting pattern analysis, rather than simply signature verification, holds promise as an effective security measure. Voice analysis relies upon each individual's speech. The unique digital signature of characteristic patterns of voice overtones can be authenticated with similar signatures already prerecorded within the computer system. Similarly, a video picture of the person attempting to access the system can be obtained and compared to previous video scan information about the individual.

Although fingerprint and handwriting analyses are well developed and exist within data systems today, voice and video discrimination are still emerging fields. Voice discrimination, however, is likely to be sufficiently developed for IOC SSDS consideration. Additional research is needed in correlating speech signatures obtained at different occasions. There must be assurance that the individual speaking is correct while maintaining flexibility that variations in speech on different days (for example, when one has a cold) will still be acceptable. Discrimination between actual voice and secretly recorded vocal messages is also necessary.

Computerized video discrimination is still being developed but promises to provide a definitive authentication of the individual.

2.3.2.3.4 Data Systems Staff Selection/Monitoring Procedures

2.3.2.3.4.1 Description. Additional security can be obtained by careful selection of and by periodic monitoring of the data system staff . Two-man policies, whereby no one individual has complete knowledge of security measures can also prove useful.

2.3.2.3.4.2 Option Categorization. Periodic staff monitoring should include systems programmers, operators, systems analysts, etc., as well as selected Space Station users. The assigning of security and privacy as an important priority within the Space Station program and the monitoring of adherence to this policy should reduce security breakdowns.

Section	Option/Issue Focus	Scope of Options/Issues Addressed and to be Addressed
2.3.2	Data System Access Controls	
2.3.2.1	Password Options	Simple: Person's name Complicated: Sentences, etc. Query System Issues: Password control Breadth of applicability Commonality among SSDS nodes Update procedures Password encryption
2.3.2.2	Data Acess Restrictions	Read vs. write access Differentiated security levels Ease of access vs. security tradeoff
2.3.2.3	Device/Personnel Controls	Specific equipment check Callback verification Personal badges, ID numbers Turn off equipment before leaving Authorized time periods Finger/hand prints Handwriting analysis Voice/video identification Staff selection/training/monitoring Two-man concept

2.3.3 OPERATING SYSTEM ACCESS RESTRICTIONS

2.3.3.1 Security Model Options

2.3.3.1.1 Description. Perhaps the key SSDS security related design issue is the degree of integrity and reliability of the operating system(s), especially if data and command communications among processors of different vendors are involved. Significant progress has been made in recent years in developing secure communications processors.

Many commercial systems are being developed to protect client data with high reliability and integrity. The Department of Defense is also actively supporting research in this area. It is anticipated that enhanced security and data protection mechanisms will continue to be implemented in new vendor hardware and associated operating systems.

A helpful reference for establishing criteria for system security has been published recently by the National Security Agency (NSA) within the Department of Defense, entitled "Department of Defense Trusted Computer System Evaluation Criteria", Computer Security Center (Fort Meade, Maryland 20755), 15 August, 1983, CSC-STD-001-83. Although military-level security is not presently required within the Space Station program, this document defines a useful framework for discussing and evaluating a broad range of security options. It provides a useful context for assessing operating system security measures in particular.

In this NSA document various classes and subclasses of protection are defined. The major classes are Minimal Protection (D), Discretionary Protection (subclasses C1, C2), Mandatory Protection (subclasses B1, B2, B3), and Verified Protection (subclasses A1, and beyond-A1). The differentiation in security protection among the four major subsets is much larger than that within the subclasses.

This document specifies trusted computer system criteria in terms of 1) security policy, 2) accountability, 3) system assurance (both operational and life-cycle), and 4) documentation. Each security class or subclass encompasses the criteria established for the next lower class or subclass, plus adds additional requirements to be satisfied.

Recently, the U.S. Air Force has established requirements for a class B2 level of interactive workstation for use at the NSTS Control Center at the Johnson Space Flight Center. NASA has also presented an assumption that "the computer assets used onboard the SS and at ground support locations will meet the class A1 requirements" (as defined in the NSA manual - see Reference 3). These recent decisions imply that the NSA requirements are an accepted NASA baseline for security definition.

Issues and options related to operating system security design are presented below.

2.3.3.1.2 Option Characterization. The primary issue is what overall system security policy will be implemented. Specifically,

1) What security, privacy, and subsystem integrity policy will be established at each node in the network? Options include the diversity or similarity of security policy among SSDS nodes, as well as the extent of implementation at each node. Will a standard A1 level of security be required throughout, or only onboard, etc.?

2) For what types of data and users? Options here include having different security policies for scientific versus commercial or command data. In particular, the high data rate scientific users may not require any encryption.

3) How extensively will the adequacy of security policy be verified? This may depend on the options selected for question 1 above. If an A1 level of security is imposed throughout the SS system, verification procedures and assurances will be well specified.

4) How will updates to the vendor and SSDS operating system(s) be managed? Options for configuration control of system updates, either by SSDS personnel or according to vendor specifications, need to be formulated. Differences in update procedures for onboard versus ground also need to be evaluated and determined.

5) Will the NSA set of criteria or a different set become the SSDS security/privacy benchmark? Although the recent NASA Kennedy Space Center SS Security Requirements Plan (Reference 3) implies an affirmative response to this question, such requirements might be modified for subsystems or particular data types if warranted.

Whatever the security criteria, the levels of security required at each network node, for each system database, and for the internode communications links need to be determined. The NSA manual focuses on individual processor security, especially of the operating system; specific criteria dealing with the communications media are not explicit.

Security Models: A widely established security model is the Bell-LaPadula model. In this model read access is allowed only to objects of a lower or equal security level; write access is determined by the *-property (star property), i.e., write access is allowed only to objects of greater or equal security level.

This model implies the implementation of a hierarchical set of security access levels. In obtaining services from different levels of the operating system, users will have varying restrictions imposed upon them. The breadth of the SSDS hierarchy of access levels among the different applications users also needs to be determined.

Different access levels to different parts of the operating system will be needed to allow systems programmers to modify system software and to permit various utility functions to be invoked. These users, however, must be restricted from modifying the secure kernel of the operating system or from utilizing it to gain access to protected system or user data. This implies a layered operating system design. Whether existing commercial operating systems can provide this protection and meet other SSDS requirements, or whether the SSDS will need a specially modified operating system is an important SSDS design decision.

Important areas to monitor are the breadth of security implementation in vendor microprocessor and minicomputer systems, and the compatibility among their security policies, especially if different vendor systems are candidates for SSDS networking. Possible performance degradation due to excessive security-related checking andmonitoring will need to be assessed.

2.3.3.2 Secure Operating System Design Features.

2.3.3.2.1 Description. Three primary concepts related to operating system security design are 1) the security monitor concept; 2) kernel control; and 3) trusted entities. Examples of existing secure communications and operating systems are presented. Optional decision areas with respect to the SSDS need to be investigated.

2.3.3.2.2 Option Characterization.

Examples of commercially available secure computer systems are recent Honeywell processors. The Honeywell Multics system has been available since 1982 and is rated as level B2 according to NSA criteria. This system has eight rings within the operating system and as a standard procedure employs encrypted passwords. The system only grants user access on a need-to-know basis and has specific mechanisms to prevent Trojan Horse or trap door types of security breaches. Many of the security mechanisms are implemented within system hardware, thereby reducing the opportunities for software modifications.

Recently, Honeywell and the Department of Defense have upgraded the multics processor to Level A1. This new Secure Communications Processor (SCOMP) is described in Reference 5. Additional A1 level projects currently under development include:

KVM/370, being developed by SDC

KSOS, by FACC and Logicon

Sacdin, by ITT and IM.BM

COS/NFE by Compion (DTI)

Gemini Computer Trusted Multiple Microcomputer Base (Level 133 design)

PSOS, by FACC and Honeywell

RAP Guard, by CSC and Sytek

2.3.3.3 Privacy of Operating System Data.

2.3.3.3.1 Description. Access control to and protection of the operating system (and its data) are basic to any computer security policy. In this subsection additional options with respect to the privacy of operating system data are explored. Included are accounting data (delineates the scope of user resource utilizations), various performance data (often contain information about the operations environment and user impacts upon it), and system data on passwords, scheduling algorithms, etc. Access control to data interchanged between the SSDS and the Technical Management and Information System (TMIS) will need to be formulated as requirements for TMIS become better defined.

Section	Option/Issue Focus	Scope of Options/Issues Addressed and to be Addressed
2.3.3	Operating System (OS) Access Restrictions	
2.3.3.1	Security Model Options	NSA Security Criteria Options Security Policy Implementations: Ground vs. space End-to-End versus each SSDS node Dependence of Data Types Verification Procedures System Update Policy Bell-LaPadula Model Option Commercial OS vs Special SSDS OS Vendor Security Policy Compatibility Performance Degradation Tradeoff
2.3.3.2	Secure OS Design Features	Honeywell Multics and Scomp examples
2.3.3.3	Privacy of OS Data	Breadth of Data TMIS Interface with SSDS

2.3.4 DATA BASE INTEGRITY/PRIVACY

2.3.4.1 Description. SSDS core and user data bases must be protected. The core data bases will contain information about available Space Station System resources and schedules; user access and priority codes to these resources; evaluation criteria to determine the validity and safety of user and System commands; etc. Each user data base, in addition to possible proprietary data, may contain data files for which access is required by the SSDS.

Maintaining the integrity of the SSDS data base(s) while allowing ongoing updates as experiments progress will be necessary. Data base update control will depend on the degree of data base centralization and on the breadth of processes (or users) for whom modification access is authorized.

2.3.4.2 Option Characterization. Important SSDS data base design options are:

1) the extent to which data base(s) are centralized or distributed.

Centralization may facilitate security implementation, but with a likely tradeoff of reduced data accessibility;

2) the control paths and procedures available for accessing and modifying the data base(s).

A single secure access and data control path should likewise facilitate security, but may reduce data accessibility;

3) the breadth of processes (or users) authorized to perform data base updates.

Multiple authorized software processes (and associated users) complicate security implementation.

A distributed data base may have multiple physical access channels within a multi-processor system. A centralized data base will usually have a unique access path. A single core data base for critical services and data with one secure physical and logical access path would facilitate security/privacy implementation and support data base integrity.

Options in three areas - 1) data base file protection, 2) statistical inference controls, and 3) data base backup methodologies will be addressed within the above contexts.

Data Base File Protection: Among the optional areas for evaluation are:

1) What hierarchical set of passwords and for which portions of a distributed (functionally or physically) data base are needed?

2) Which users can access which subsets of functions, services, or data?

3) Should physically separate data bases (or portions thereof) be implemented to facilitate security/privacy?

Some data may have a natural allocation between space and ground. Other data, such as commands, probably require centralized verification and restricted accessibility to provide an adequate level of integrity and security/privacy.

4) Which SSDS functions and services will be distributed and sometimes duplicated within different nodes?

5) What degree of operating system security (as defined by NSA) will be implemented at each SSDS node?

These decisions and the breadth of encryption utilization within the SSDS will depend on

1) the degree of perceived threats to the SSDS (a Space Station System threat assessment is in process (See reference 3));

2) the functional allocation decisions resulting from other tradeoff studies;

3) the available vendor DBMS software support; and

4) the types of hardware/software subsystems eventually proposed within the SSDS.

Security/privacy concerns and impacts on and by these decisions will therefore require a stepwise and iterative approach.

Section	Option/Issue Focus	Scope of Options/Issues Addressed and to be Addressed
2.3.4	Data Base Integrity/Privacy	Integrity vs. Need for Updates Issues: Centralized vs. Distributed Access Control Paths Breadth of Authorization for Updates Optional Areas: Data Base File Protection Password Hierarchy User Access Physical Separation Distributed/Duplicated OS Security at each Node Factors: Risk Assessment Other Trade Studies Vendor Software Adequacy SSDS Design Decisions

2.3.5 DATA ENCRYPTION TECHNIQUES/POLICIES

2.3.5.1 Scope of Applicability/Survey of Techniques

2.3.5.1.1 Description. Data encryption is a technique of concealing the information content of a set of digital data by transforming it into a new set by means of a precisely defined computational algorithm. The efficacy of the resultant encryption security is based on the practical impossibility, from a computational standpoint, of reversing the process (called decryption) in a practical amount of time without knowledge of the decryption "key". The encryption algorithm may utilize pairs or sets of large prime numbers in performing a predetermined set of computational steps. The reverse or decryption process requires knowledge of the prime number keys and the algorithmic steps. Often the algorithm is not kept secret; the security rests on the computational difficulty of deciphering or guessing the "key" prime number components. The encryption process renders the data set unintelligible to anyone not knowing the decryption key. The ultimate efficacy of the encryption process usually rests on the safeguarding of the algorithmic "keys".

There are two types of encryption algorithms commonly used. The first uses the same "key" for encryption and decryption. This is referred to as a private key encryption algorithm, since the "key" must be kept secret throughout the process. The protection of keys and the dissemination of information about key changes can be difficult to secure.

The second algorithm type uses different keys for encryption and decryption. Only one of the keys in this case needs to be protected. This approach has two advantages -

1) messages using a public encryption key can be transmitted that only the receiver can interpret with his private decryption key; and

2) the authenticity of the sender can be verified since he can encrypt his signature with a message using a private key, but any receiver can decipher his signature using a public decryption key.

By this process, the sender's signature can be guaranteed (and he can be held accountable for his data transmission), since everyone can verify his accuracy using the public decryption key.

2.3.5.1.2 Option Characterization. The leading private key encryption process is the Data Encryption Standard (DES) formulated by the National Bureau of Standards (NBS) in 1977 (see Reference 6). The DES technique uses a well documented algorithm and a 64 bit binary key (56 bits are used by the algorithm and 8 bits for error detection). With 70 quadrillion permutations of the 56 bits, deciphering the message using trial and error techniques is tedious.

The most widely disseminated public-key encryption method is that proposed in 1978 by Rivest, Shamir, and Adleman (the RSA technique) of M.I.T. The authors of this technique claim that "the efficacy of their decryption may be faster since multiprecision arithmetic operations are simpler to implement than complicated bit manipulations." The DES scheme is probably faster, however, if specialized hardware chips are used.

The Knapsack Approach:

Alternative public key encryption schemes based onthe "knapsack" approach have been recently shown to be easily decipherable. The knapsack cryptosystems are based on the difficulty of deciphering the sets of integers used to add up to a given large integer. At the Crypto '84 meeting last summer in Santa Barbara, California, Ernest Brickell of Sandia Laboratories presented results negating the security provided by "iterated knapsacks."

Among the quotes relevant to encryption that were made at Crypto '84 were:

"I think the breaking of iterated knapsacks is quite surprising and indicates a degree of insecurity that had not been suspected at all"
(Ralph Merkle, of ELXSI)

"You have to be extremely cautious when claiming that a cryptosystem is strong. The history of cryptography is essentially a history of failures. Lots of cryptosystems,, proved to be insecure, sometimes with disastrous results." (Adi Shamir, Weizmann Institute)

"The most intriguing question is whether you can develop proof techniques that will show the security of cryptosystems. If you could do this, it would be the biggest breakthrough in cryptography because at last you would be able to show that concrete cryptosystems just will not be broken in the future unless there is a certain amount of time." (A. Shamir)

The above illustrate that most encryption algorithms are not failsafe, but their security rests on the inability of anyone to decipher them to date within a reasonable amount of time.

Bulk Encryption Devices:

Bulk encryption devices, many utilizing specialized hardware microchips, are now available which can meet the needs of most Space Station users. Many devices exist commercially that can encrypt in the MBPS range; these devices utilize microchips which can keep up with data rates in this range at no additional system overhead. Devices with capabilities in the 100 MBPS range are coming into existence.

Encryption, besides applications for highly proprietary commercial data (and potential national security data), may be most applicable to operating system (OS) protections and secure commanding.

With respect to OS protections, encryption can be applied to limit access to command processes and critical system data, protect passwords and/or data file names, conceal resource scheduling algorithms and data, etc.

With respect to secure commanding, restricted and constrained uplink commands to the Space Station (or another SSPE) will require checking to insure that they do not jeopardize the System safety or health. A dual key encryption approach may be especially valuable if both the authenticity of the sender and the integrity of the command content can be simultaneously protected. Command acknowledgements (encrypted if necessary), in addition to sender-receiver authentications, provide additional security assurance before implementing restricted (or constrained) commands.

Restricted commands will normally be evaluated and so categorized well before any mission. Their security might be enhanced if they were tightly bound to the operating system, with its secure kernel maintaining the privacy of the keys. Encryption might simplify necessary command checking since it could provide a higher level of assurance that the command is legitimate and accurate. The availability of rapid encryption/decryption subsystems combined with highly secure keys could facilitate the desire for both near real-time commands and commanding 'transparency.'

Although the present NASA perspective, as reflected in the GSFC Customer Requirements document, is to give the responsibility for encrypting data to the user, this philosophy may need reevaluation with respect to restricted command handling. The tradeoffs between commanding transparency versus command validation need to be resolved satisfactorily. Space Station system safety must be assured while the user maintains flexibility and commanding transparency.

The efficiency of the DES and RSA encryption algorithms, with respect to security assurance, appears adequate at present. Attempts to foil these protective schemes needs constant monitoring, however. More efficient computational algorithms, and/or faster encryption microchips (e.g., NSA developments) require continued investigation with respect to their applicability to the ultra-high data rate experiments proposed.

The DES encryption technique appears easily implementable within the SSDS based on current technology. One option is whether to incorporate DES as the standard SSDS encryption technique, or have it as one of many. The scope of applicability of encryption needs to be decided. This decision will be impacted by

1) general NASA policies;

2) risk analyses of the envisioned SSDS threats;

3) the evaluated security of vendor operating systems being considered; and

4) continued analyses of commercial and foreign proprietary concerns as they evolve.

2.3.5.2 Options for Interfacing Customer Encryption Equipment

2.3.5.2.1 Description. The "SSIS/SCS shall provide a standard interface to customer supplied encryption and decryption equipment." (Reference 1, 5.4.4). The acceptance of an encryption standard, e.g. DES, within the SSDS, might support this goal. The question to be resolved is whether a single standard, such as DES, is judged to be adequate to satisfy the necessary security requirements of the Space Station Program. If the SSDS were to provide an optional encryption service, the selection of a single standard such as DES might be desirable. One factor in evaluating the adequacy of DES (or any other standard) is that for highly critical procedures or data, double DES encryption could be performed. If the 56 DES bit field were judged inadequate protection, double encryption using two distinct keys would likely be adequate.

Section	Option/Issue Focus	Scope of Options/Issues Addressed and to be Addressed
2.3.5	Data Encryption Techniques/ Policies	
2.3.5.1	Scope of Applicability/ Survey of Techniques	Private key vs. public key algorithms DES vs. RSA vs. alternative standards Areas of Applicability: Commercial proprietary data Command and Control Operating System Access User Responsibilities Adequacy of Vendor Security Systems Risk Analyses
2.3.5.2	Interface to Customer Encryption Equipment	To Be Determined

2.3.6 Physical Security Design Options. SSDS physical security involves protection of the hardware, software, and internode communication links. Adequate alarm notification is also needed whenever a security violation is detected.

2.3.6.1 Hardware

2.3.6.1.1 Description. Four areas relevant to hardware protection are: 1) sabotage prevention; 2) reliability/fault tolerance; 3) power failures; and 4) the spares management policy.

2.3.6.1.2 Option Characterization.

Sabotage: Unfortunately, sabotage is an ever present threat in modern life. As stated in Reference 3, "In the design of computer access controls, each user of the computer assets and the information data base must consider all other users hostile" (Assumption 3.3.3). Hopefully, the personnel authentication options presented in section 2.3.2.3, as well as other system checks, will minimize sabotage. Sabotage on a broad scale, i.e., bomb threats, is probably beyond the scope of the Space Station Data System (or Space Station Program) to address fully. Personnel screening and strict adherence to security policy will be necessary.

Reliability/Fault Tolerance: The SSDS must be a reliable and fault tolerant system. Subsystem reliability/maintainability/availability (R/M/A) is a key criterion in the computer subsystem selection.

Important SSDS options involve the 1) reliability and 2) interchangeability of the system components within and amongst network nodes. Critical onboard and ground systems will require backups, at least functionally. At IOC, many functions will be performed on the ground which will eventually migrate to particular Space Station elements. Insofar as possible, compatibility and interchangeability among backup hardware components amongst the different nodes are desirable. The greater the compatibility and backup redundancy, the better the overall system reliability and security. Additional emphasis on these topics is discussed in other areas of the SSDS Options list.

Power Failures: Power failures, especially onboard, can threaten the health and safety of the crew, onboard specialists, and/or critical missions. Communications could be interrupted by power failures. Alternate power systems will be an integral part of the Space Station element designs. The SSDS will also need access to these alternative power sources in the case of failure. Some systems (UNIVAC mainframes, etc.) have rotating flywheels to maintain power in case of a failure. Qualified flight processors of this type may not be suitable, however, for onboard consideration. Although alternate power sources are not a primary responsibility of the SSDS designers, the SSDS needs to be compatible with the alternative power sources implemented elsewhere within the Space Station program.

Maintenance/Spares Management: In addition to backups and alternate power assurance, periodic online maintenance/monitoring of SSDS subsystems will be necessary. Important options are: 1) the scope of the periodic maintenance and 2) the types of indicators for needed maintenance. With what regularity will periodic maintenance be implemented and for which subsystems? Who will be responsible for interpreting malfunctions and what alternative remedies will be provided? Will vendor supplied maintenance diagnostics be adequate? Many of these issues cannot be answered until subsystem design decisions are made, but the importance of maintenance and monitoring need to be emphasized.

An intelligent management policy for SSDS spare parts is necessary. Since Shuttle flights to replenish onboard components will be weeks apart, system components crucial to the Station health and safety either must be provided in sufficient quantity onboard or alternative subsystems must assume such critical operations. Backup components may prove impractical for particularly heavy components such as power transformers, magnetic tape units, etc. More limited backup systems (in the event of failure) with reduced capability but providing basic functioning are options to be considered.

The spares management policy must utilize effective configuration management whereby up-to-date inventories of spare parts are available via interactive (terminal) access to crew and ground personnel. Alternative designs of the SSDS configuration management functions need to be evaluated.

2.3.6.2 Software

2.3.6.2.1 Description Protection of SSDS software against either sabotage or accidental harm is another important security issue.

2.3.6.2.2 Option Characterization. Options related to the security of the operating system have been described in Section 2.3.3. Utility programs and important core and user standard software services also need regular backups.

All software backup and modification procedures need to be verified, maintained, and proven effective against destruction. Crucial ancillary or engineering data likewise need to be backed up. Options defining the scope and mechanisms for protecting and updating software and the data within the SSDS are addressed in other areas of the Options List.

2.3.6.3 Communications Links

2.3.6.3.1 Description. An additional discussion of the availability and performance of the SSDS communication links, addressed briefly in Section 2.3.1.4, is presented next.

2.3.6.3.2 Option Categorization. Protecting the security of the communication links will require coordination between the Space Station program and other institutional facilities within which the SSDS operates. For example, the TDRSS network is a shared facility. Security constraints desired by the Space Station program will need to be coordinated with the representatives responsible for maintaining this network. Similarly, the SSDS design must take into account that assuring TDRSS network performance and providing its maintenance are not Space Station program responsibilities. Options for coordinating with these institutional representatives and for assuring adequacy of service need to be addressed.

In addition to restricting access to the information transmitted among these links one must also protect against possible data modification or denial of service, as mentioned in Section 2.3.1.4.

To address possible data modification within the SSDS network, information communicated between nodes can be encrypted. Some header information, however, may need to remain in the clear to identify the destination of the message, its sender, and whether (if a command) it is restricted, constrained, or unrestricted. Acknowledgements from the receiver back to the sender will be necessary; the receiver identification can also be encrypted within the message. As an alternative, information identifying the command as restricted, constrained, or unrestricted can be duplicated within an encrypted portion of the command. In this manner the command accuracy can be checked, but information in the clear can be used to facilitate its transmission.

In addition to data modification, extraneous data generated from other sources and simulated to look like actual network data or commands need to be reliably identified and discarded. Including time generation information within the encrypted portion of a message can discriminate it from a retransmission of a copied older message.

The final area discussed deals with communication link availability and possible denial of service. The health and safety of the Space Station could be jeopardized if communications are impacted which involve emergency procedures being transmitted from the ground, for example. One SSDS design option is to provide alternative pathways for all vital communications, i.e., an emergency, direct-ground-to-Space-Station communication link in the event of TDRSS unavailability. Such an alternative link can have reduced bandwidth if it only handles commands. Similarly, alternate ground communication pathways are needed for transmitting critical SSDS data among nodes.

2.3.6.4 Alarm Responses

2.3.6.4.1 Description. Whenever a maintenance or a security violation problem is encountered, optional SSDS alarm responses and procedures are possible.

2.3.6.4.2 Option Characterization. One option is the extent of centralization (physically or functionally) in dealing with alarms and security problems. Detection of problems will occur throughout the system; violations affecting Station safety and health, for example, will need to be communicated to the Space Station Operations Control Center and to the crew, and perhaps elsewhere.

Under what alarm conditions will there be automatic shut down of subsystems or functions, or automatic failover? The scope of checking and responding to security alarms can be very extensive, or minimal. Besides health and safety related violations, to what extent should SSDS security/privacy violations be noted? Should the SSDS automatically respond to these situations or should it simply send a message to an operations control center operator or to a crew member? The degree of automated response could be limited to disseminating information about the problem, and follow up actions could be manual. Operator evaluation of alarms might reduce the need for large automated systems with significant software development costs to address problems which may infrequently or never occur, or which are not dangerous to the Station or crew.

Section	Option/Issue Focus	Scope of Options/Issues Addressed and to be Addressed
2.3.6	Physical Security Design	
2.3.6.1	Hardware	1) Sabotage Control 2) Reliability/Fault Tolerance Issues Component Interchangeability Emergency Backups 3) Power Failures Alternative Systems Emergency Procedures 4) Spares Management Exact duplicates vs. alternatives Configuration Management
2.3.6.2	Software	Backup Procedures Update Procedures Protections
2.3.6.3	Communication Links	Coordination with Institutional Facilities Use of Encryption Elimination of Extraneous Data Emergency Procedures Alternative Communication Links
2.3.6.4	Alarm Responses	Centralized vs. Distributed Automatic shutdown criteria Breadth of Monitoring Automatic vs. Manual (Operator)

2.3.7 Network Monitoring Policies. Described herein are techniques to monitor data system transactions for the purpose of observing unusual activity. In this manner, attempted breaches of security sometimes can be discovered. Four areas will be addressed -

1) monitoring the types and number of accesses to the data system and its associated data bases;

2) monitoring the amount and type of data transmissions across internode communication links;

3) compiling statistics on CPU utilization; and

4) maintaining transaction logs.

2.3.7.1 Data Base/Data System Accesses

2.3.7.1.1 Description. Monitoring the accesses and the access pattern to the data system and its data bases can often lead to information about security breaches. Oftentimes an individual who attempts to gain unauthorized access may not know the required passwords or procedures. He may make repeated access attempts and/or use the computer itself to cycle quickly through various passwords or data base file access codes. The monitoring of access attempts and the alarming of a security officer when a large number of accesses has occurred is an important security measure.

2.3.7.1.2 Option Characterization. A primary option is the breadth of data base and system access for which monitoring shall be implemented. The extent of logging normal accesses to the system and to each data base also needs to be determined. For accesses which fall within the system security guidelines, to what extent should data transfers and data modifications be logged for potential analysis if necessary? The periodicity of reports to the user of all attempts to access, transfer data from, or modify his data files are important. For what threshold levels of excessive system access should security alarms be initiated?

Access checks are normally implemented by means of tightly controlled software within the operating system. Extensive access checks, however, can potentially impact the performance and responsiveness of the operating system.

2.3.7.2 Data Transmission Monitoring

2.3.7.2.1 Description. Protecting the integrity of the communication links has already been discussed in Sections 2.3.1.4 and 2.3.6.3. Communication and transmission path security can be assisted by monitoring of transactions across and between communication links. In the same manner that accesses to the data system and data bases can identify questionable activities, remote system access requires transmission across a communication link. The monitoring of repeated, excessive transmissions can provide an additional system alert. In this case the individual or individuals responsible for multi-node network security must be alerted. The periodicity and types of transactions monitored and logs reported need to be determined.

2.3.7.3 CPU Usage.

2.3.7.3.1 Description. The monitoring of excessive CPU usage can also pinpoint security breaches. For example, if an unauthorized user does gain access to the system, he may use the computer resources to attempt to decipher encrypted passwords. This decryption process may require significant CPU usage. The monitoring and reporting of unusual CPU usage can alert a security officer to this situation.

2.3.7.3.2 Option Characterization. The extent and regularity to which excessive CPU usage is flagged and to whom this information is to be sent are SSDS design options. The monitoring and reporting of normal (or slightly above normal) CPU utilization may impose unnecessary operating system overhead. The limits beyond which excessive CPU usage should be flagged and the breadth of system processes monitored are additional options. The availability of vendor supplied monitor and performance assessment tools may impact the scope of such implementation.

2.3.7.4 Transaction Logs/Archival

2.3.7.4.1 Description. The transaction pattern among different users and different network nodes, i.e., which user processes and user identifications communicate, can provide an important record in analyzing security breaches. The monitoring and recording of all such transactions, however, is impractical. Immediate alarms should be initiated, however, if illegal transactions are detected, ie., if data were sent from a higher to a lower security level, etc.

2.3.7.4.2 Option Characterization. Important options are 1) the extent of transaction logging, 2) the length of time such logs are maintained, and 3) for whom hard copies should be periodically produced. The need to check for irregularities in transactions other than those already implemented within the system security model may be minimal if an A-1 level of system security is implemented.

Section	Option/Issue Focus	Scope of Options/Issues Addressed and to be Addressed
2.3.7	Network Monitoring Policies	
2.3.7.1	Data Base/Data System Access Monitoring	Breadth of Applicability Logging Procedures Alarm Thresholds Potential Performance Impacts
2.3.7.2	Data Transmission Monitoring	Multi-node Monitoring Periodicity and
Breadth		of Reports
2.3.7.3	CPU Usage	Extent/Regularity of Monitoring Adequacy of Vendor Supplied Software
2.3.7.4	Transaction Logs/Archival	Internode Transaction Pattern Extent and
Periodicity		of Logging

REFERENCES

1. Goddard Space Flight Center Space Station Office, Customer Requirements for Standard Services from the Space Station Information System (SSIS), Revision 1, September 19, 1984.

2. Department of Defense Computer Security Center, Department of Defense Trusted Computer System Evaluation Criteria, Fort George G. Meade, Maryland, August 15, 1983.

3. John F. Kennedy Space Center, Space Station Security Requirements Plan, Basic Issue, July, 1984.

4. Honeywell, Multics Data Security, Honeywell Information Systems, Inc., 1982.

5. "Computer Security Technology, Preventing Unauthorized Access," Computer, Volume 16, Number 7, July, 1983.

6. "Data Encryption Standard," Federal Information Processing Standards Publication 46, January 15, 1977.

7. R.L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, Volume 21, Number 2, February, 1978, pp. 120-126.

8. Martin, James, "Software, Techniques, and Architecture," Computer Networks and Distributed Processing, Prentice-Hall Inc., Englewood Cliffs, N.J., 1981, pp. 517-541.

9. D.W. Davies, D.L.A. Barber, W.L. Price and C.M. Solomonides, Computer Networks and Their Protocols, John Wiley & Sons, New York, pp. 390-415.

10. Peterson, Ivars, "The Unpacking of a Knapsack, Science News, Volume 126, November 24, 1984

11. White House National Security Directive, no. 145, "National

2.4 TIME MANAGEMENT

Time is the most common basis for initiating actions and correlating events in all aspects of the Space Station, not just in the data system. The SSDS will have a direct involvement in three major categories: time reference; time distribution; and time tagging of data. A number of specific problems arise in each category. Most of these problems have been addressed and solved on previous NASA programs, while some are introduced or made worse by special requirements of the Space Station.

2.4.1 Time Reference

Local time sources throughout the SSDS must be related to a primary standard. The purposes are to provide a common reference time for scheduling future events, initiating current events, and correlating past events both within the SSDS and throughout the entire Space Station community. The ground systems in NASA have proven techniques to reference local time sources to a standard such as WWV. Since the movement of the Space Station prevents the same ready access through the onboard system, another reference source must be provided.

2.4.1.1 Onboard Reference Source

2.4.1.1.1 Description

The three options for the onboard time reference are an onboard master timing unit, the global positioning system, and the onboard computer oscillators. The need for a time reference during buildup may influence the choice and is discussed separately.

2.4.1.1.2 Option Characterization

2.4.1.1.2.1 Master Timing Unit

An atomic clock with a low drift rate may be used as the onboard master timing unit (MTU). This is an established technology with drift rates on the order of 1 part per billion over 24 hours, which is 86.4 microseconds per day or 2.6 milliseconds per month (reference 1, section 2.4.1.1.4.). Both the ground and onboard systems interact with the MTU. The ground has the responsibility to monitor the magnitude of the MTU offset from a primary source such as WWV, and to command updates as needed. Any such updates must be only for small amounts (see 2.4.3.1 for reasons), on the order of 1 to 5 milliseconds. The general ground procedure to determine the amount of offset is to monitor a very precise point at which the telemetry system reads time through a direct connection to the MTU, such as the start of a master frame every 2 seconds. The delays are known from this point through the MTU, through signal propagation to the ground station, through early ground recording, and to a point at which the ground system reads its own time reference. The difference between the onboard time and the ground time, after allowing for the known delays, is the offset needed for a time update command to the onboard system.

The same general procedure would be used for the Space Station, although exact details will depend on the final equipment. The end result is that an MTU onboard the Space Station can be used as a time reference with only minimal ground support for updates about once per week. The technology and techniques are already established. The unique problem of initializing the MTU is discussed in section 2.4.1.3.

2.4.1.1.2.2 Global Positioning System

The global positioning system (GPS, reference 2 of section 2.4.1.1.4) is likely to be the navigation reference for the Space Station. GPS includes a time source in the receiver onboard the Space Station that provides time accuracy better than 1 microsecond. The option therefore exists to use GPS for both the navigation reference and the time reference.

The GPS provides a time reference that does not depend on NASA ground support, a desirable feature for autonomy of the Space Station. The problems of initialization and correction of clock drift are a GPS responsibility, so that the NASA can assume a very accurate and driftfree clock.

2.4.1.1.2.3 Computer Oscillator

The local oscillator of a computer can be used as an onboard time reference. This method is used internal to each computer for control of its own functions, such as a flight control computation 10 times per second or a once per second thermocouple sample. In order to prevent all of the computers from drifting relative to each other, one can be selected as the time reference to which all other computers equalize their times (see 2.4.2).

This method is not practical for any but emergency situations. The problem is that the oscillators are crystals, not atomic clocks, and have drift rates that are unacceptable for long term references. Controlled thermal and voltage conditions can keep the drift rate below 10 parts per million, or about 0.9 seconds per day. The ground would need to be very active in providing frequent small updates to prevent the error from building up to the extent that an update would noticeably affect the system operation (see 2.4.3.1). Even at an update rate of once per orbit (about 5400 seconds for low earth orbits) and a drift rate of 1 part per million, the individual updates would be 5.4 milliseconds each.

This technique is not recommended as the primary onboard reference, but should be considered as the next-to-last faultdown step from failures of the preferred sources. The last step is for all computers to use their own local oscillators without equalizing to a single computer due to a major failure, causing all computers to drift independently either faster or slower. In either case, resetting all times to match the reference after repairs is likely to be a significant perturbation to the routine operation of the network.

2.4.1.1.2.4 Buildup

The decision of which onboard time reference source to install must consider the buildup of the Space Station. The reference configuration builds the structure over a series of 7 flights. The earliest flights may not have a GPS processor and antennas installed, so that option might not be applicable to the early phases. The alternative may be an MTU on the early flights, with a later switch to GPS, but this option requires more development effort than choosing either the GPS or an MTU as the permanent source. The alternative of using just the internal computer oscillators may also be acceptable. The resultant time drift will result in attitude errors relative to the desired local vertical or gravity gradient attitude, and the frequent ground updates will perturb some functions at the update. However, this time accuracy may be adequate for later docking for installation of the GPS or MTU equipment.

2.4.1.1.3 Projected Capabilities

The GPS and MTU reference sources of today are adequate for Space Station in 1987 and beyond.

2.4.1.1.4 References

1) Master Timing Unit, document MC456-0051, by Rockwell International.

2) Global Positioning System, papers published in Navigation, 1980, by The Institute of Navigation.

2.4.1.2 Ground Reference Source

The ground is expected to use WWV as its time reference to prevent drift of the ground processing clocks from a standard time. Synchronization techniques that have been developed and used for NASA centers for other programs should be transferable to the Space Station program.

2.4.1.3 Initialization

2.4.1.3.1 Description

Both the time reference source and the computers will require initialization to the correct time. The details depend somewhat on the onboard reference and on the state of the onboard reference. The most involved case has an MTU as the reference source, with neither the MTU nor processors having a valid time. This situation is normal at first power on of the system, which may occur before launch into orbit or after arrival on orbit. A GPS reference source may eliminate the steps which initializes the time reference. If the computer oscillator is the reference source, then the steps terminate after the computer is initialized.

2.4.1.3.2 Option Characterization

An MTU update is assumed to be based on an implementation that says "When your clock reads ABC, the actual time is XYZ." The time ABC is the coincident time, and the time XYZ is the replacement value to be used. The MTU is also assumed to accept a reset command which sets the MTU to a predefined value, such as midnight of January first. The alternative that says "The time right now is XYZ" is not very satisfactory, since uncertain delays between the computer issuing the update and the arrival at the MTU would degrade the accuracy of the initialization. Another satisfactory alternative is an implementation which tells the MTU to "adjust the current time by amount XYZ" (either forward or backward), with only minor changes to the general procedure.

When the first computer in the network is turned on, part of the initialization will read (at least, attempt to read) the MTU. If the MTU is not accessible or the value fails some tests for reasonableness (e.g., seconds are from 0 to 59, hours are from 1 to 12 or 24) the computer can set its internal time to some convenient arbitrary time, such as midnight on January first (time 0 of day 1). An operator can then command the computer to synchronize the MTU time to the computer time, which might be implemented by resetting the MTU to time 0, computing the next coincident time from this known reset state, and sending the coincident time and related replacement time to the MTU. The computer and MTU will be synchronized, but not at the correct time, after the coincident time is reached, and should remain together through the normal time distribution (see 2.4.2).

At this point the computer and MTU are together, either from an initial reasonable reading being accepted from the MTU or from the above procedure. No further action is required if the time is acceptably accurate, as would occur for any computer turned on after the MTU was initialized or for a time source such as GPS which is not initialized by the Space Station.

If there is no MTU or if the MTU must be set to an initial time (an operator decision), the operator will command an incremental time update by the error in the MTU value. The implementation would use the incremental time update procedure used for operator commands of small updates (a few milliseconds) at infrequent intervals during normal operations. The computer would determine the appropriate coincident time and related replacement time value, issue the command to the MTU, and also adjust its own internal time at the coincident time to continue to match the MTU. For reasons given in section 2.4.3.1, the large initial update should also cause the computer to recycle as though the computer had just been turned on.

Both the computer and MTU are initialized and synchronized at the end of this procedure. Time distribution (see 2.4.2) will keep the two together, barring some kind of failure. Later computers will pick up the correct time from the MTU as they are turned on. If the MTU fails and a replacement has to be initialized, the "set MTU equal to computer time" command will make the MTU match the computer, after which the operator can command an incremental time update to remove any time offset from oscillator drift while the MTU was off.

2.4.1.3.3 Projected Capabilities

Specific details may change during the development of the Space Station, with options to trade automation versus operator intervention at some places. At this point in the project, the procedure is well enough understood and proven on other projects that no further initialization options or trades are needed for the current architecture study.

2.4.1.3.4 References

Space Shuttle Program Orbiter Project, Computer Program Development Specification, Volume 1, Book 7, STS System Level Requirements, Software. NASA JSC document SS-P-0002-170. Sections 4.6.2.4 and 5.9.2 describe time management.

2.4.2 Time Distribution

Time distribution is concerned with propagating the time from the reference source (MTU, GPS, or one of the processors) through the entire network of processors, devices, and interface devices (IDs). The object is to maintain an acceptably small offset from the reference at each point. The required accuracy is 1 millisecond (reference of section 2.4.2.1.4). Time distribution contributions to this error of less than 100 microseconds should be attainable with the hardware/software approach described below.

A basic problem is that the reference source is not directly accessible to every user in the network. The only direct connection is to the ID attached to the reference source, and possibly to a few special subsystems such as telemetry to the ground. All other users must access time indirectly, but without unpredictable delays through the data network and processors. The general approach to time distribution is to provide many secondary time reference sources by a three step procedure. First, the ID attached to the reference source will be equalized to match the reference source. Second, this ID will transmit its time value to all of the other IDs in its local area network, including any bridges or gateways to other onboard networks, which allows these nodes to equalize their times to match the received data. Third, each device or processor attached to an ID can access time from that ID in order to equalize the processor's time to the ID. Some degradation is expected as the time goes through successive levels of remoteness from the reference source, but the hardware and software option described in the next three sections should provide excellent time accuracy relative to the time reference.

2.4.2.1 Reference Source to Its ID

2.4.2.1.1 Description

The first step of time distribution is to get time from the reference source to the attached ID. A basic assumption is that the NIU has some processing capability and includes a clock of about one microsecond quantization which can be both read and written. One procedure is used if the time reference source has a register which can be accessed directly by the ID (the GPS or an MTU). Another procedure is required for high accuracy if the source is a computer without direct access to the time register by an ID.

2.4.2.1.2 Option Characterization

If the time reference source is the GPS or an MTU, the ID will be able to give a "read current time" command to that unit and get an immediate response. The ID will read its own internal time as closely as possible to the point of this command. The difference between the value from the ID and the value from the reference source can be adjusted by any fixed delays, and the result used to correct the ID internal timer. For safety, a limit test should be used to protect against failures.

If the time reference source is a computer, this procedure may have an unacceptable accuracy. The problem is that the command to "read current time" may have unpredictable and large delays within the computer, notably if the computer is currently busy with some other task and does not immediately respond to the command. The command will probably be implemented as a high priority interrupt to the computer. The response will be for the computer to read its own internal time, to adjust for known delays in accepting the interrupt, and to send the result to the ID. An alternative at this point is to leave the data in memory to be read by the ID using the direct memory access (DMA) protocol. The limitation on accuracy is the unknown random delays in the computer.

A more accurate transfer of time from the computer to the ID requires a clock register which can be written into by the computer and read by the ID. The computer would periodically write the low order part of its timer into this register to prevent any noticeable drift, and the register would continue to count from whatever value was loaded. The ID could read this value periodically to update its own internal clock. The register should probably be a part of the ID and be driven by the ID oscillator, in order to keep non-standard hardware (from the viewpoint of the processor) outside of the processors. A 16 bit register at one microsecond quantization is adequate for this function. (See also 2.4.2.3 for another use of this register.)

2.4.2.1.3 Projected Capabilities

Time can be matched between the time reference source and the ID to within a few microseconds by designing the interface to allow direct access to the clock in the reference source by the ID. This accuracy, available with existing technologies, is adequate for the Space Station project.

2.4.2.1.4 References

Customer Requirements for Standard Services from the Space Station Information System (SSIS), GSFC, 9-17-84.

2.4.2.2 To IDs in Onboard LAN

2.4.2.2.1 Description

The second step of time distribution is to transmit time from the ID attached at the reference source to the other IDs, bridges, and gateways on the same local area network. The general concept is for the source ID to read its time, create a ID-to-ID data packet, and transmit this packet to the other IDs. Each ID which receives this packet would read its own internal time, difference this value with the time in the packet, and apply the difference as a correction to its own internal time. The difficulty is that several delays are involved in this procedure. While some of the delays are known and can be included in the adjustment, some residual unpredictable delays are likely to be above an acceptable level. The two largest uncertainties are likely to be gaining access to the transmission media after reading time at the sending end, and responding to the receipt of the message in order to read time at the receiving end.

2.4.2.2.2 Option Characterization

Special hardware support in the ID is needed to avoid these delays. Instead of having the processor in the ID read its clock, the ID data transmission hardware should read the clock at the actual start of transmission and store this clock value in the message. Then, the receipt of the last bit of the message should cause the hardware in the receiving ID to read its own internal clock and save this clock as a part of the message buffer. Each of these clock values is at a precisely defined point during the message transmission. The time to transmit the message is known from the transmission rate and delays through any intervening nodes (significant for a token ring, at least). The difference is therefore well defined, and can be used to equalize the clock in the receiving ID to the clock in the transmitting ID.

Note that the above technique for time tagging messages arriving at a ID is very useful for tracking messages through the network during development of the network, for performance measurements during checkout and verification, and for problem analysis of an operational network. The technique would be recommended even if not required for accurate time distribution.

2.4.2.2.3 Projected Capabilities

Time can be distributed between IDs in a LAN with an error much less than 1 millisecond by designing the message transmission hardware to timetag each message at the actual sending and receiving points. These capabilities are available today, and satisfy Space Station requirements over the life of the program.

2.4.2.2.4 References

None.

2.4.2.3 To Processors or Devices in LAN

2.4.2.3.1 Description

The third step of time distribution is to get time from the ID to the attached processor or device. This transfer is controlled by the processor, not the ID, to avoid any problems which might result if the ID updated the processor's time at an uncontrolled point during computations.

2.4.2.3.2 Option Characterization

A simple option for the processor to get the time from the ID is for the processor to send a "read time" request to the ID through the same mechanism used to perform normal input/output requests. The ID would respond by reading its own internal clock and returning the value to the processor. This method has the disadvantage that the exact ID processing time is likely to be unpredictable because the ID may be busy with some other activity at the point of the request from the processor.

A second option is to provide a direct path from the processor to the hardware clock in the ID, with the intent of avoiding these unpredictable delays. The processor would read both its own clock and the ID clock as close together as possible, compensate for any known delays, and then adjust the processor's clock for the difference between the two values. This method has very good accuracy. A direct path is likely to be available between the ID and the processor to allow the processor to sense the status of the ID, so that this direct path to the ID's clock should be very little extra hardware. The direct path has the advantage of simple time management software in the processor. The method is useable if the ID has at least three programmable clocks. (One clock is equalized to the reference source, which jumps at each update. A second clock is independent of the updates, as needed for measuring intervals. The third clock generates interrupts to initiate scheduled events within the processor.)

If only two clocks are available (one for interrupts, one for other uses), then two of these function must be combined. A proven method (Space Shuttle) is to combine a free-running hardware clock with a software clock. Each adjustment to time is implemented by adding the change to the software part of the combination without affecting the hardware part. The sum of the two parts is therefore equalized to the reference. By avoiding any jumps in the hardware clock at time adjustments, this method lets the hardware clock be used for measuring short intervals without unwanted perturbations. For example, assume one wished to determine the time from the start to the end of a procedure by reading the hardware clock at the start and end. This kind of measurement should not include adjustments which might have been made to the clock during the interval to account for equalization updates, leap seconds, or other discontinuities. The problem is that the combined parts are not directly available to the attached processor as desired, since the processor will not be able to directly access the software part of the clock.

The third option requires only two clocks in the ID, at the expense of some added complexity in both the processor and ID software. All changes to time are added into the software part of the clock, as above, while the hardware part is unaffected by changes. The hardware part can be read directly by the attached processor, and the processor can also ask for the complete time from the ID. The procedure at time equalization of the processor is then as follows. First, the processor reads its own internal time and the ID hardware register as close together as possible. Next, the processor requests the ID to send to the processor the ID time together with the hardware clock value that is part of that time. The processor can combine these four pieces of data (processor time and related ID clock, ID time and related ID clock), with any known delays in reading related sets, to obtain the difference between the processor and ID clocks. The difference is used to adjust the processor's clock to match the ID.

The first option may be acceptable, but the accuracy of time in the processor then has an undesirable dependence on unpredictable delays in the ID. The second and third options provide high independence between the ID and the processor, while giving very good time accuracy. The choice between the second and third options is based on the number of programmable clocks within the ID.

2.4.2.3.3 Projected Capabilities

Time in the processor can be matched within a few microseconds of the time in the ID by designing the interface to allow the processor to have direct access to the clock in the ID. This capability, using existing technology, is adequate for the Space Station.

2.4.2.3.4 References

None.

2.4.3 Time Tagging

Time tagging is the generic name for associating an event with the time of occurrence of that event. The time may be in the past ("Reboost occurred at 9:03 A.M."), at the present ("The time is 1:48 P.M."), or in the future ("Call at 3:30 P.M."). Several major problems of time management are related to time tags such as the above, with various applications requiring different kinds of times. The problem areas are discontinuities in time, accuracy of time tags, and formats of time tags. The reference configuration (section 2.4.3.1.4) requires the network operating system (NOS) to timetag all subsystem messages.

2.4.3.1 Discontinuities

2.4.3.1.1 Description

The fundamental visible time used on the Space Station will be Greenwich Mean Time (GMT), measured in seconds from midnight on January first. This system avoids frequent discontinuities to which people are accustomed (seconds and minutes jumping from 59 back to zero, hours jumping from 12 (or 23) back to 1 (or 0), daylight saving time, time zones). All of these discontinuities create difficulties in programming real-time tasks. However, GMT itself still has several discontinuities. In order of decreasing magnitude, these are the year-end rollover (from day 365), the leap year (a variation of the year-end rollover), and the leap second. In addition, the equalization of each ID and processor to the reference will create small (microseconds or milliseconds) local discontinuities at the update cycle, typically once per second.

The SSIS, of which the SSDS is a subset, must consider year—end rollover and leap seconds throughout the design, since these effects are very difficult and expensive to add afterward. The leap second has been especially difficult to handle. Reference 1 of section 2.4.3.1.4 tried to add processing to Space Shuttle ground and onboard systems to accommodate a flight across a June/July leap second boundary. The conclusion was that the only practical way to transition all systems across the leap second was to turn off all processors (both onboard and on the ground at JSC) and reinitialize in flight! The technique currently adopted has JSC suppress locally any leap second for the duration of a mission, and to compensate all range tracking data for the one second offset that results. Even ignoring the confusion of having two different sets of data for the same time tag (or two different time tags for the same data), this method is clearly not acceptable over the flight duration of the Space Station.

A basic problem of time management is that some applications cannot tolerate the discontinuities which are necessary to other applications. Some examples will illustrate the needs, at various levels of tolerance.

2.4.3.1.2 Option Characteristics

2.4.3.1.2.1 Fine Timing

Assume that some application has the need to delay a very short period of time, such as 200 microseconds. This need might occur in sending a pulse to some external device by a sequence which turns a signal on, waits 200 to 300 microseconds, and then turns the signal off. Even these short delays should be programmed by reading a hardware clock, computing the clock value at the end of the delay period, and issuing the signal until the clock value exceeds the computed end value. If a time equalization update of 150 microseconds were to occur during this delay, and the update was effected by changing the hardware clock used to time the delay, then the signal would be maintained for an incorrect amount of time. The software should not implement the obvious alternative of delay loops based on instruction execution time, even for such short intervals. To do so makes the software dependent on the speed of a particular processor, and requires that the software be modified to run on a newer processor for a technology upgrade.

2.4.3.1.2.2 Timeout Tolerances

The previous example may be unrealistic, because such short delays are nearly always embedded in the operating system and executed with interrupts disabled, preventing the equalization update during the delay. However, a slightly larger discontinuity can affect many applications. Assume that an application has asked for some data from an external device and will wait up to 1 second before declaring an error and initiating some recovery action. This delay is typical of the desired response to crew requests at a workstation. Most data acquisition which does not directly involve a person will have timeouts that are much less than 1 second. If a leap second were to occur during this interval, and affect the timer being used for the timeout, then an error may be reported because no actual time has elapsed (1 second timeout minus 1 second leap second adjustment).

2.4.3.1.2.3 Calculations

Assume that some application needs to determine the rate of change of a measurement by sampling a sensor every two seconds, and then dividing the difference in sensor values by the difference in sensor time tags. If a leap second were to occur between two readings and be included in the time tags, then the computed result would be too high or too low, depending on the sign of the update (either 1 second or 3 seconds, but not the real value of 2 seconds).

Comparison of ground tracking to onboard navigation has been a difficult problem across a leap second. Because of numerous examples like the above, spread over many applications, the onboard software is compelled to use a time that has no discontinuities (other than the equalization updates to prevent long term drift). Yet, the ground tracking system is synchronized to GMT and must accept the leap second update. At an orbital rate of 5 miles per second in low earth orbit, the sudden apparent jump due to the time discontinuity may trigger a navigation update to the onboard system which creates a real error! This situation will also exist with GPS.

2.4.3.1.2.4 Periodic Processing

Many applications have a need for high rate periodic processing. Flight control is a typical application, with a possible requirement to execute ten times per second at precisely spaced intervals of 0.1 seconds. Periodic processing is initiated by the operating system when time exceeds the next scheduled time of execution for a process. This next scheduled time is computed, either by the application or by the operating system, by adding the repetition interval (0.1 seconds in this example) to the current time of execution, and inserting this value in order in a list of timed events for the processor. The time until the item at the top of this list is loaded into a clock to generate an interrupt at the required time. The processor is then free to execute application programs until the interrupt occurs. A leap second has a drastic effect on high rate periodic processes, if the leap second is included in the computation of the time until the interrupt. The effect will be either that the processor immediately tries to execute 10 cycles all at once (a multiple cycle overrun) if time appeared to jump forward, or does not process any cycles at all for a full second (which might, for instance, leave control jets firing much too long). Neither situation is desirable, and the analysis to prove that the effects are benign can be very expensive.

A slightly different periodic processing example requires the inclusion of the discontinuity at a leap second. Assume that some monitor function is intended to execute once per minute at 5 seconds after the minute occurs. Now the periodic processing is tied to an external clock, unlike the above paragraph. If the leap second is ignored in this case, then the many leap seconds during the life of the Space Station would cause the processing to drift far away from the 5-seconds-after point as seen by an outside observer. The primary point is that both of these apparently similar users have different real requirements that should be accommodated.

2.4.3.1.2.5 Earth-based Observations

Most of the earlier examples show that discontinuities are undesirable. But the discontinuities were actually introduced to satisfy the needs of other applications. The basis of the need is the slowing rate of rotation of the earth. Without the adjustment of the leap second, the Greenwich meridian would drift relative to solar noon, so that the sun would not be directly overhead at 12:00 noon GMT. The slowing effect also influences stellar observations. Users who need earth-based observations require the jump of the leap second to keep a close correspondence between the solar noon and 12:00 noon clock time.

2.4.3.1.2.6 Summary of Discontinuities

The above examples concentrate on the problems caused by a leap second, because that has proved to be the most difficult case. The small jumps due to equalization to a time reference are generally of interest only to the lower levels of the operating system for precision timing of very short intervals. These cases can be handled either by deferring the adjustment until after the interval, or providing a clock which is unaffected by the adjustments. The very large discontinuities at year-end rollover (including leap year) are so large as to be readily detected and treated. The usual treatment is to keep a time (ephemeris time) within the system that "just keeps going" and never resets or jumps for year-end rollover or leap seconds. The frequent equalization adjustments to prevent long term drift relative to a reference source are of little interest to most users above the operating system level. Ephemeris time is used for scheduling high rate periodic processes and for time tags which enter into computations such as navigation or sensor rates of change.

2.4.3.1.2.7 Options for Handling Discontinuities

The basic time used within the network should be an ephemeris time that has no discontinuities. This is necessary to support the many functions that need a continuously increasing time. One second of ephemeris time always equals one second of International Atomic Time (TAI).

The reference epoch for this ephemeris time should be January first of either 1950 (Earth Mean Equator 1950, EME-50) or the year of launch of the Space Station. The former is an international standard reference. The latter has the advantage of exactly matching GMT until the first discontinuity. Either reference can hold time in a 64-bit computer word quantized to 1 microsecond over hundreds of years, either as a long integer or as 8/56 floating point (8 bits of exponent and 56 bits of fraction).

Conversions between this ephemeris time and several widely used times should be provided by the operating system as a common service needed by many users. The particular time systems will evolve as Space Station requirements mature. As a minimum, the conversion to Universal Coordinated Time (UTC) and GMT are required, both of which include leap seconds. Conversions for manual display will be needed, and are candidates for common operating system services. These would typically be conversion from the internal form of GMT to days, hours, minutes, seconds and possibly fractions of a second. Section 2.4.3.3 discusses other potential standard formats for communication uses.

2.4.3.1.3 Projected Capabilities

Adequate time tag capabilities exist today. The objective of this sections is to emphasize that the data system must be designed from the start with the recognition that at least two fundamental time bases are required. The "visible" system will be GMT, simply because that is the most widely used time in the world. However, the inherent discontinuities cause intolerable problems for many applications. Therefore, the "internal" system will be a continuous time base, probably EME-50 because of its wide use. The software system should make both available to all applications, and provide the translation between the two.

2.4.3.1.4 References

1) Series of memoranda, FS52/G. L. Richeson to FS52/Head, Trajectory Logic & Processing Section, JSC, early 1982. Subject is leap seconds that occur while a Space Shuttle flight is in a mission, and multiple Shuttles with mixed times before and after a leap second. Quote: "It was presented that both Onboard and MOC (Mission Operations Center at JSC) would have to be brought down and be reinitialized if the leap second has to be incorporated inflight."

2) Space Station Reference Configuration Description, JSC-19989, August 1984.

2.4.3.2 Accuracy

A user of time generally has two concerns about the measured value. The first is the accuracy, which defines how far away the measured value is from some reference source. The time distribution methods of section 2.4.2 which use special hardware to precisely timetag the start and end of input/output message transfers should be able to propagate the reference time throughout all IDs to an accuracy better than 100 microseconds. Those methods which involve the ID software in message timetags will be less accurate because of unpredictable delays in processing, perhaps allowing errors of 1 to 5 milliseconds. As an estimate of the effects of accuracy, in low earth orbit in a local vertical or gravity gradient attitude, 1 millisecond contributes a dowrange error of about 25 feet and an angular error of about 0.25 arc-seconds.

The second concern is quantization, or how fine time may be resolved. One second is the usual quantization for manual interaction, but this level is not adequate for internal computer purposes. Many functions within a processor take place in less than 1 millisecond. There is also an interaction with the time accuracy if the quantization is too coarse. For example, if time is quantized to 1 second, then the above accuracy of 100 microseconds cannot even be approached. The IDs and processors should have the ability to read and update their internal times at a 1 microsecond level in order to avoid the expense of analyzing how coarse time can be and still be useable to every application and to the operating systems.

The above discussion is for accuracy and quantization of "absolute" time. Some applications have a need for much better values than can be met by distribution of time over a network. These applications must either have a direct connection to the reference source or have an internal source. Two examples are typical. Very early time accuracy requirements (about 1971) showed a need for 0.1 microsecond accuracy in data time tags in one subsystem on Space Shuttle. The MTU has a precision oscillator (1 part per billion), but a time quantization of 125 microseconds, which clearly does not meet the apparent need. However, the actual requirements were for very precise interval measurements, with no concern about GMT, which could be attained by using either the MTU or an internal oscillator as a frequency source combined with a counter to generate the needed interval measurement. The second example is the GPS, which requires time accuracies near one nanosecond to measure speed-of-light delays. Even a 1 microsecond accuracy is inadequate here, since light would move 1000 feet in that time, while the intended position error is much less. GPS supplies its own precision atomic clock.

2.4.3.3 Standard Formats

2.4.3.3.1 Description

The wide variety of formats used for time causes serious problems in space missions. One user may desire time in hours, minutes and seconds in BCD (a internal machine format, binary-coded decimal). Another may want time in seconds and fractional seconds in the EME-50 reference. Still another may require time in integer microseconds, notably for internal computer timing. International atomic time (TAI) does not include leap seconds, while GMT or UTC do include this effect. The conversion between formats requires both program development effort and computer execution time.

2.4.3.3.2 Option Characterization

The NASA data systems standards program addresses standard time formats as a method to manage the number of variations, notably for data communications (reference of section 2.4.3.2.4). The second is defined as the TAI second, with an option for UTC. The epoch may be either EME-50 or an unspecified epoch. The fundamental unit is integer seconds and fractional seconds. The integer may have one to four octets (8 bits, commonly called a byte) of capacity, and the fraction may have one to three octets. The maximum format can therefore handle times up to 135 years at a quantization of about 60 nanoseconds, covering any Space Station data system needs. Each time word carries its own descriptor so that the user can determine the format during processing, with no need for prior agreements.

2.4.3.3.3 Projected Capabilities

Time standards adequate for Space Station communication and internal uses will evolve by 1987. The two fundamental bases will probably be International Atomic Time (TAI) and Universal Coordinated Time (UTC), both for the EME-50 epoch as time zero. TAI provides the continuous time needed by many users, and UTC provides leap second adjustments. GMT is readily extracted from UTC by ignoring multiples of 1 day.

2.4.3.3.4 References

The New Standard Spacecraft Timecode, by Edward B. Connell, NASA GSFC, (undated, probably mid-1983).

2.5 COMMUNICATIONS

2.5.1 Space Communications

The organization of the Space Station/SSDS communications traffic, both external and internal, is influenced by architectural, format, protocol, and scheduling factors in addition to existing equipment characteristics. Considering these factors a preliminary list of options have been developed. These options have been arbitrarily partitioned into transmission characteristics and architectural options.

2.5.1.1 External Architectural Options

The following list of options have been generated considering the factors enumerated above and the current technology level and that projected for the 1990's.

1. 1, 2, 3, 4 ball TDRSS configurations.
2. TDRSS augmented by enhancements or TDAS.
3. TDRSS augmented by a commercial satellite utilizing a combination of TDRSS and ACTS technology.
4. TDRSS augmented by direct downlink to the DSN.
5. All users required to provide for their requirements in excess of TDRSS capacity.
6. TDRSS capacity increased by advanced modulation techniques.

These options are characterized in Table 2.5.1-1.

2.5.1.2 Internal Architectural Options

The internal data flow has been partitioned into audio/video and data channels. The reason for this type of partition is due to the unique characteristics and the volume of video data. An additional constraint is that both audio and video have analog transmission options which are not generally available for data transmission. This does not eliminate a digital option for audio and video. In addition to the factors enumerated above the factors of bus loading, overhead penalty, and data criticality must be considered.

PRECEDING PAGE BLANK NOT FILMED

The following options have been generated:

1. All data transmitted on parallel on board SSDS busses utilizing a packetized format.
2. Parallel data buses with different characteristics.
   - Core data bus utilizing a packetized format for low rate transmission.
   - User data bus using virtual connections for high rate/bulk data transmission.
   - Direct memory access data bus for bulk transfer of stored data or bulk uploads.
3. Another option would combine data, voice, video, in a digital format on the same bus structure; however, the very high rates which would be required would require some major technology improvements.

These options are characterized in Table 2.5.1-2.

2.5.1.3 Downlink Transmission Options

Utilizing the factors enumerated above in concert with the known characteristics of the data to be transmitted the following options have been generated. These options generally offer choices between various format, protocol, and scheduling alternatives.

1. Time Multiplex Schemes
   a. Dedicated time slots for high rate or bulk data (virtual connection) with the remaining allocated time for low rate packetized data.
   b. Dynamic allocation for high rate or bulk data with the remaining time allocated for low rate packetized data.
   c. A totally packetized and multiplexed scheme.

2. Channel Allocation Scheme No. 1

- o S-band channel reserved for core data
- o Ku-band I channel reserved for high rate or bulk data.
- o Ku-band Q channel reserved for other experimental data.

3. Channel Allocation Scheme No. 2

- o S-band channel reserved for core data
- o Ku-band dynamically allocated.

4. Channel Allocation Scheme No. 3

- o Ku-band for all transmission
- o S-band reserved to smooth peak loads.

These options are characterized in Table 2.5.1-3.

2.5.1.4 Uplink Transmission Options

The characteristics, volume, and relative criticality of uplinked communications traffic is now well defined. Thus considering these limitations and utilizing the general factors enumerated above the following preliminary option list has been generated.

1. Channel Allocation Scheme No. 1

- o All core commands/uploads on S-band.
- o All user commands/uploads video on Ku-band.

2. Channel Allocation Scheme No. 2

- o All uplink on Ku-band
- o S-band for overflow

The effect of program uploads on channel occupancy may suggest an adaptive channel allocation for these upload periods. These options are characterized in Table 2.5.1-4.

Table 2.5.1-1

EXTERNAL ARCHITECTURAL CHARACTERISTICS

1. 1, 2, 3, 4 BALL CONFIGURATION

Advantages

- Increased system capacity
- Increased availability to SSPE's

Disadvantages

- Increased management complexity
- Necessity for additional ground terminals
- Necessity for increased terrestrial capacity

2. TDRSS AUGMENTED BY ENHANCEMENT OR TDAS

Advantages

- Increased system capacity
- Increased available to SSDE's
- Possible inter-satellite networking

Disadvantages

- Increased management complexity
- Necessity for additional ground terminals or ground terminal modification
- Necessity for increased terrestrial capacity
- Physical, mechanical, electrical limitations on present configuration

3. TDRSS AUGMENTED BY A COMMERCIAL SATELLITE UTILIZING A COMBINATION OF TDRSS AND ACTS TECHNOLOGY

Advantages

- Increased availability to SSDE's
- Increased system capacity
- Geographical distribution of ground terminals relieving congestion of terrestrial links

Disadvantages

- Less reliance on long haul terrestrial network and reduced delivery delay
- Increased management complexity
- Necessity for additional ground terminals
- Support for dual relay satellite systems
- Technology still being investigated.

4. TDRSS AUGMENTED BY DIRECT DOWNLINK TO DSN

Advantages

- Increased capacity
- Utilization of existing ground stations
- Dual communications paths for safety and reliability
- Worldwide geographical distribution of ground terminals

Disadvantages

- Limited link capacity
- Limited connect time

5. ALL USERS REQUIRED TO PROVIDE FOR THEIR REQUIREMENTS IN EXCESS OF TDRSS CAPACITY

Advantages

- Limit communication support to well defined facilities
- Encourage user consideration of alternatives to massive raw data dump

Disadvantages

- Shift communications responsibility to users

6. TDRSS CAPACITY INCREASED BY ADVANCED MODULATION TECHNIQUES

Advantages

- Increased capacity
- Stable TDRSS space and ground segments

Disadvantages

- Requires redesign of transponder modulators and/or support of dual modulation capability

Table 2.5.1-2

INTERNAL ARCHITECTURAL CHARACTERISTICS

1. All data transmitted on parallel SSDS buses utilizing packetized format.

   Advantages

   - Standardized formats and protocols

   Disadvantages

   - High overhead penalty
   - Transmission subject to unscheduled interruption from higher priority traffic
   - Transmission network subject to being loaded down by a few high volume users
   - Poor utilization of traffic capacity

2. Parallel Data Buses with Different Characteristics

   Advantages

   - Segregation of traffic
   - Increased utilization of bus traffic capacity
   - Reduced overhead penalties
   - Increase access to low priority - low volume users

   Disadvantages

   - Non-standardized formats and protocols

Table 2.5.1-3

DOWNLINK TRAFFIC CHARACTERISTICS

1. Time Multiplexed Schemes
   a. Dedicated

      Advantages
      - Tailored to data characteristics
      - Simple data handling procedures
      - Well defined data boundaries

      Disadvantages
      - Possible poor utilization of channel capacity
      - Poor flexibility for contingencies or growth

   b. Dynamic

      Advantages
      - Efficient utilization of channel capacity
      - Inherent flexibility for contingencies or growth

      Disadvantages
      - Complex data handling procedures

   c. A Totality

      Advantages
      - Standardized protocols and formats

      Disadvantages
      - High overhead penalty

2. Channel Allocation #1

   Advantages
   - Well defined data boundaries
   - Simple data handling procedures

   Disadvantages
   - Possible poor channel capacity utilization
   - Poor flexibility for contingency or growth

3. Channel Allocation #2

Advantages

- Well defined data boundaries
- Data handling procedures are moderately complex
- Moderate to good channel capacity utilization
- Inherent flexibility for contingencies or growth

Disadvantages

4. Channel Allocation #3

Advantages

- Inherent flexibility
- Good channel capacity utilization

Disadvantages

- Complex data handling procedures

Table 2.5.1-4

UPLINK TRANSMISSION CHARACTERISTICS

1. Channel Allocation Scheme #4

   Advantages

   - Well defined data boundaries
   - Moderate data handling complexity
   - Inherent flexibility for contingencies or growth

   Disadvantages

2. Channel Allocation Scheme #2

   Advantages

   - Inherent flexibility
   - Good channel capacity utilization

   Disadvantages

   - Complex data handling procedures

2.5.2 WIDE AREA COMMUNICATION/PROCESSING OPTIONS

2.5.2.1 OVERVIEW

This paper describes the options for Wide Area Communications/Processing. Wide area communications is that part of the Space Station Information System (SSIS) that includes the communication and processing of Space Station data that occurs from its reception on the earth as, for example, at White Sands through its delivery at the customer locations. This subject, in effect, represents a global network involving both terrestrial and space links. It will include several nodes (Space Station Program Elements):

- Primary Earth Terminals (initial contact with Space Station);
- Space Station Operations Control Center (SSOCC);
- Polar Oriting Platform Control Centers (POPCC);
- Co-Orbiting Platform Control Centers (COPCC);
- Payload Operational Control Center(s) (POCC);
- Orbiting Maneuverable Vehicle (OMV) Control Center;
- Orbiting Transfer Vehicle (OTV) Control Center;
- Regional Data Centers (RDC);
- Data Handling Centers (DHC); and
- Customer facilities.

Some of these nodes, such as the RDC's and POCC's, may perform the same functions. The allocation of functions identified in Task 1 is in many cases a trade-off between space SSPE's and these ground nodes, as well as between ground nodes.

This network will carry diverse traffic with many different quality level requirements supporting many different customers. It will evolve in size, capacity, level of service, and technology throughout the lifetime of the Space Station Program. This network will accommodate specific customer messages with data rates ranging from hundreds to millions of bits per second. It may use packet, message and/or circuit switching while using terrestrial point-to-point links and/or satellite broadcast modes. There will be from one to three separate data streams (Space Station, COP and POP) from space to earth, each of which will contain multiplexed data, elements of which will have different end destinations, quality, privacy, delivery and timeliness requirements. Superimposed on this variety is the requirement that the constitution of each data stream will change as the missions change. The cost of reconfiguration of the network is this a paramount criterion.

This data flow is characterized as originating from one to three space data sources and delivered to a multitude of destinations. Conversely, in the uplink direction, commands, data, video and audio signals will originate in multiple points on the globe and arrive at the Space Station usually in multiplexed streams. Consequently, the wide area network will be complex and characterized by many options.

A number of stringent and unique requirements are imposed on this network. These include: operational transparency, customer knowledge of data status, incorporation of ancillary data in the customer data stream, privacy and security, real time response, up to 300 Mbps on some links, data integrity not to be less than 10^{-9} bit error rate on some links. Some of these requirements are contradictory and must be balanced. For example, customer knowledge of data status and network privacy can be contradictory as well as the requirements for real time response with high data rates, that will, of necessity, impose storage and therefore delays.

2.5.2.2 OPTIONS

This section contains a brief description of a variety of options that exist for this network design. Some options will be programmatic, as for example, the location and authority associated with decisions on data flow paths and priorities. Most options will be technical in nature.

2.5.2.2.1 NETWORK CONFIGURATIONS

One of the primary set of options that will impact the final implemention of the wide area communications is the choice of network configuration. This set of options include:

- a number of primary earth reception stations;
- access from the primary earth reception stations to the wide area network;
- connectivity between primary earth stations and other regional sites (nodes);
- the number of regional sites (nodes);
- the allocation of functions between nodes; and
- customer access to the network.

For the purposes of this document, a primary reception station is defined as a terminal that first receives data from the Space Station.

One proposed configuration has two earth reception stations, referred to as NASA Ground Terminals (NGTs). They consist of the antennae, reception equipment, data capture equipment, multiplexers and formatting equipment for retransmission to such nodes as JSC or GSFC.

At IOC it is planned to have at least two single access TDRS channels available, which will require two primary reception stations to receive data simultaneously from each of the two operational TDRS spacecraft. With respect to data processing and subsequent transmission on the network, two options exist for this configuration; one primary earth station can transfer all received data directly to the other by means of a fiber optic link or, alternately, all processing and preparation for transmission can be performed at both stations. In the latter case, each station could independently transfer data to the network. Each station could have Space Station, COP and POP data segments to place on the network at any given time. Depending on the network design, this will require knowledge of the data destination of each segment. One means of providing this knowledge is to imbed the destination in each data element on the Space Station. In this case, the telemetry protocol must contain space for the data destination.

Furthermore, in the growth versions of the space station, additional primary reception stations could be implemented as was indicated by the TDAS studies which identified five reception stations. This type of configuration offers several options for controlling data flow through the Wide Area Network. One would involve highly complex scheduling such that the network would maintain knowledge of which regional reception station should receive which data and at what time each station would be visible to the Space Station from a TDRS satellite. This knowledge could be maintained in the Space Station Data Management System which would then control traffic to each earth reception station or, ultimately, by a Network Control Center. Another option would have the ultimate destination imbedded in the data stream and transferred as appropriate thru the destination. Direct space-to-ground links other than TDRS should they be employed, will impact the Wide Area Communications topology. Links that would include Space Station to a technologically improved geostationary spacecraft (such as ACTS) to ground. Such links would bring the customer closer to the payload to provide for direct interactive control. The connectivity between the primary earth stations and the RDC's represents various topological options. This topology will include customers located at various destinations on the planet - a global network. Thus, some primary earth stations could be located outside the United States. Similarly, RDC's and POCC's could be located outside of the United States.

Connectivity options include: providing communication links between each primary earth station and all POCC's and RDC's or providing such links to only certain dedicated POCC's and RDC's. In the latter case, the space station would have to sort data in terms of the primary earth station location, a design option which seems desirable for RDC's and/or POCC's are located in foreign areas such as Europe. A trade-off between space station data handling complexity and transmission costs is implied. A major parameter of this trade-off is the traffic volume to each RDC.

2.5.2.2.2 MULTIPLEXING/DEMULTIPLEXING/PROCESSING LOCATION

Both forward and return data streams will require demultiplexing, multiplexing, and processing prior to and after receipt by/from the customer. The location of these processes has several options and consequently cost and timeliness impacts on the data system performance.

The data stream(s) received from the Space Station will consist of multiple data channels with different levels of data security/privacy, quality requirements and destinations. This stream(s) must be processed. Some of the associated options are:

- separation by customer identification at the initial point of reception from space at a data handling center, at a single regional data center (such as GSFC), or at multiple regional data centers (or combinations of each); and

- likewise Level 0 processing at a single location or multiple locations;

It should be noted that in addition to demultiplexing/multiplexing and Level 0 processing, typically each node should check for errors

and add some form of error detection prior to transmission to the next node. Depending on the final configuration, each node may also be required to read a header in order to direct the specific data stream to the proper location.

The network may be designed as a message or a packet switching network. In this case, some level of demultiplexing will be necessary at the initial point of reception in order to recognize the packets or messages to allow switching to occur. Demultiplexing could be performed at the primary earth station and at a regional data center prior to delivery to the customer. Message and/or packet switching networks will require that each packet or block be reformatted to accommodate switching as required by the network. Thus, the protocol (standardization) and network design are interdependent.

Similarly, the Level 0 processing could be performed at either a single location such as the initial ground station (and thereby routed to some customers directly) or performed at a single final destination prior to customer delivery (or, again, a mixture of these nodes).

Customer data locations will definitely affect the topology of the communications network if total cost is taken into consideration.

2.5.2.2.3 TRANSMISSION MEDIA

Various network transmission media may be employed, such as radio frequency, fiber optics, microwave, or standard hard-wired. In this context radio frequency refers to terrestrial-to- DOMSAT-to- terrestrial links. A separate White Paper, section, Transmission Media, discusses the technical options for transmitting data. This section emphasizes transmission options based on types of service with less emphasis on technology potential. The choice of media type will depend upon data traffic, current technology, cost, security requirements, and standardization constraints. For example, fiber optics may replace DOMSAT links for certain data rates and certain links. Some transmission media options are:

- frequency of transmission;

- media type;

- the choice of commercial services; and

- o the choice between commercial services and NASA networks; NASA can currently handle high data rate traffic from White Sands over

two 50 Mb/s transponders to other NASA nodes such as JSC, GSFC and KSC. These transponders make use of leased commercial links. Another network referred to as the NASA Integrated System Data Network (NISDN) is being implemented which services fourteen nodes with a total capacity between all nodes of 15 Mb/s. NISDN can be expanded to double the 15 Mb/s simply by frequency of transponder hopping. Further expansion may be achieved by using 60 Mb/s transponders, a technology that will shortly exist; this capacity may be doubled also by frequency or transponder hopping. By the mid to late 1990's, the 60 Mb/s will likely be increased by at least a factor of two.

As stated previously the NISDN network will initially be designed to carry traffic to 14 nodes; theoretically, each of these nodes, could be used to distribute customer data on customer specific links.

NASA is developing another system for administrative data traffic, known as the Program Support Communication Network (PSCN), that will also service fourteen nodes, one of which is White Sands. This network, is capable of supporting video and voice teleconferencing, the former at frame rates compressed to 1.544 Mb/s. The use of this PSCN network for some Space Station traffic represents another option.

Most of this traffic will likely be over leased commercial lines. As such, the technology employed in these links will be determined by the commercial carriers and not the SSIS. The transmission over satellite links would include options such as single channel per carrier versus Time Division Multiple Access (TDMA). Currently, the choice between lease lines is determined by competitive government procurement procedures. In effect, the choice and type of network links employed are determined by available commercial links based on cost and technology.

Fiber optics is increasingly becoming competitive with satellites as a mode of long-range data transmission. Currently, several companies are actively expanding fiber optic networks across the United States and internationally by means of undersea optical cables. For instance, AT&T recently stated that it will spend at least $2 billion over the next two years in deploying a 21,000 mile network of fiber optic cabling. Other companies, such as MCI and Southern Pacific, are also actively installing long-range fiber optic networks. Over 25 separate companies are in the process of implementing long-haul fiber optic links throughout the United States. International long-haul fiber optic links are also being developed for international traffic: a 7,200 nautical mile cable from California to Japan is being developed by a consortium of 22 companies including AT&T. This cable will be capable of transmitting 37,800 simultaneous conversations.

Advantages of fiber optics relative to conventional RF transmission include: isolation (minimum grounding, shock and lightning interference), EMI free transmission (minimum interference, cross- talk, electro-magnetic radiation enhanced security), significant signal bandwidth (currently operating at 400 Mb/s with near term growth expected to 1.2 Gb/s) and lower cost. Current cabling costs range from $.50 per conductor meter up to $1.00 per conductor meter; a cost which is projected to decline significantly. Other costs are:

individual splices at $20 - $40, with mass splices reduced to $5 to $10 each, individual connectors range from $20 to $100 each. Installation cost, however, would represent the largest cost factor in a fiber optic channel. Given the relative sensitivity of cost data to the competing carriers, no attempt has been made in this study to determine an actual cost per bandwidth mile for long-haul fiber optic communication. However, in comparison, a typical DOMSAT link consisting of 14 earth stations and a transponder bandwidth of 32 Mbps would lease for approximately $220 thousand per month excluding installation and termination cost. Nevertheless, existing cost for both service types will likely be reduced as prices to the users due to an over capacity that will likely result from current installation activity.

Given that fiber optic links may be employed in wide area communication networks,several technology options exist which include:

- wavelength - 0.85 microns to 1.6 microns;
- transmitter - injector laser diode (ILD) versus light emitting diode (LED);
- receiver - PIN versus avalanche photo diode (APD); and
- fiber - single mode versus multimode;

The current technology trend is toward the longer wavelengths (1.6 microns) using multimode transmissions. Bit error rates of 10 $^{-9}$ over long haul distances are being achieved with repeater distances of approximately 200 kilometers. In addition, by use of wavelength, division multiplexing (WDM) up to ten channels of different wavelengths may be conducted in a single fiber. High data rates (in excess of 150 Mbps will employ long wavelength LED's (1100 to 1600 microns).

2.5.2.2.4 NETWORK CONTROL CENTER CONFIGURATION

The NASCOM network data transmission is scheduled by the Network Control Center (NCC) and reconfigured by NASCOM circuit switching. The Space Station data loads and data handling requirements will require an NCC upgrade with multiple associated options, among which are:

- degree of control centralization;
- degree of diagnostic centralization;
- if decentralized, location of control nodes;
- failure recovery techniques;
- control priorities;
- degree of automatic reconfiguration;
- type of status presentation; and
- types of data base design;

The network could be designed so that a central control center monitors all data flow paths and processing, including control and all data redirection. The centralized control center must maintain real time status and must have decision autonomy in consonance with the timing of all operations. To a large extent these operations can be predetermined - scheduled - but unexpected failures or mission changes will require real time control. High data rates will tax any implementation of a centralized control center.

An option is for decentralized control with routing determined at remote gateway locations based on current gateway traffic and link data queues. Decentralized control must have distributed loading throughout the system; other options include the amount of control "reach" at each of these nodes (the amount of network knowledge at any given node).

Diagnostics may also be centralized or decentralized. In the former case, all status and control commands would be centralized at a control center. Conversely, the diagnostics could be located at remote portions of the network and recovery effected remotely. Failure recovery offers other options ranging in degree from fully automatic to fully human controlled. An example of the latter would be the case where the bit error rate (BER) is monitored on all links at a central facility. An operator upon seeing degradation beyond acceptable limits would reconfigure the links. An option is to have automatic line reconfiguration.
Given catastrophic failure or even some specified level of degradation, the network must remain in an operatonal state. Data recovery could be prioritized; first in/first out (FIFO), by customer priority level, or by network efficiency priority level. An example of the latter case would be to initiate recovery at the node with the greatest backup queue.

Various options are associated with the type and locations of network operational displays. All displays may be centralized at a single console characteristic of central facility control. Other options are to locate displays at Regional Data Centers or at any node which requires multinode control and status information. Since customers require knowledge of the operational status, in particular the status of the data as it flows through the system, distributed control and status might be an imposed network requirement. In addition to the location of this information, there are many options associated with the level and quality of display. The level of display can range from reasonably simple statistics to elaborate graphics that provide pictorial summaries of operational status. Further, the type of display may vary from location to location depending on the degree of operational control that can be exercised at any particular station.

The network control center will utilize one or more data bases which will indicate current schedule, operational status, projected network use, failure history, etc. Read and write access to these data bases would be based on a "need to know." Thus, there exists options relative to data base access.

Other options associated with network control centers include: the frequency of polling gateways for configuration and diagnostic status; the techniques for presenting alarms to operators; the strategy for preparing alarm reports (electronic or paper); the strategy for network planning; and the availability of utilities for analyzing network downtimes. The type and scope of pre-operation and operational simulations of network evolution represent another set of options.

2.5.2.2.5 STANDARDIZATION

Section 3.1 describes the standardization options for the SSDS. The options are intimately intertwined with the design of the Wide Area Communications Network and are, therefore, applicable to the same. The important standards that will affect the Wide Area Communications are the LAN and communications protocol. No doubt, some form of packets will be used to transport data between SSPE's. These packets will require some standardized protocol. The extent that these packets must be converted during transport to and from space will clearly impact the processing requirements of the Wide Area Communications. Further, data transport within the Wide Area Communication Network will require protocol to allow for error-free transmission from node to node. Different ISO headers for each wide or local area subnetwork may be needed as illustrated in Figure 2.5.2-1. The topology is such that some modes includes customer facilities while others include NASA facilities then different protocols might be employed if for no other reason than to accommodate a general desire for commercial standard interface to customers. The existing NASCOM block is highly inefficient for data transport as was discussed in the Standard's paper. In order to accommodate a circuit switching or packet switching network, it may be necessary to employ protocols that allow for automatic traffic redirection at any given mode. It is clear that the network topology and the protocols and standards used for communications are intimately related.

Relative to the ISO standards, the important layers that will impact the network design are the transport, network and link layers. The definition of any standards for these layers will significantly effect the network characteristics. The choice of standards for these layers, while interdependent with topology requirements, will determine the proper requirements required in the network.

The transport layer selected basically specifies the data protocol for end-to-end communications. No adequate telemetry protocol presently exists for customer payload-to-payload control processing that provides for the data transfer either from one mission (experiment) to many customers or for many missions multiplexed into a single data stream to many customers. The choice of a transport protocol will substantially establish the form of the end-to-end data system and certainly the Wide Area Communications Network.

The network layer provides for the protocol to support network-port to network-port data transfers, e.g., from onboard BIU interface to any similar such interfaces (whether in space or on the ground). Again, this protocol, several of which exist, must be imbedded in the transport layer protocol to provide for efficient data stripping and manipulation with minimum overhead and machine complexity. In some designs, this network layer may be selected to be a LAN compatible protocol, and as such, may differ from the space segment protocol and may even differ within the Wide Area Communication network in order to accommodate different LAN's including customer facilities. Similarly, the data link layer protocol should be imbedded in the overall standard protocol in a manner that minimizes overhead within a maximum error free transmission. Given the anticipated high data rate and variation in data rates, the choice protocol may differ for the data associated with the different data rates.

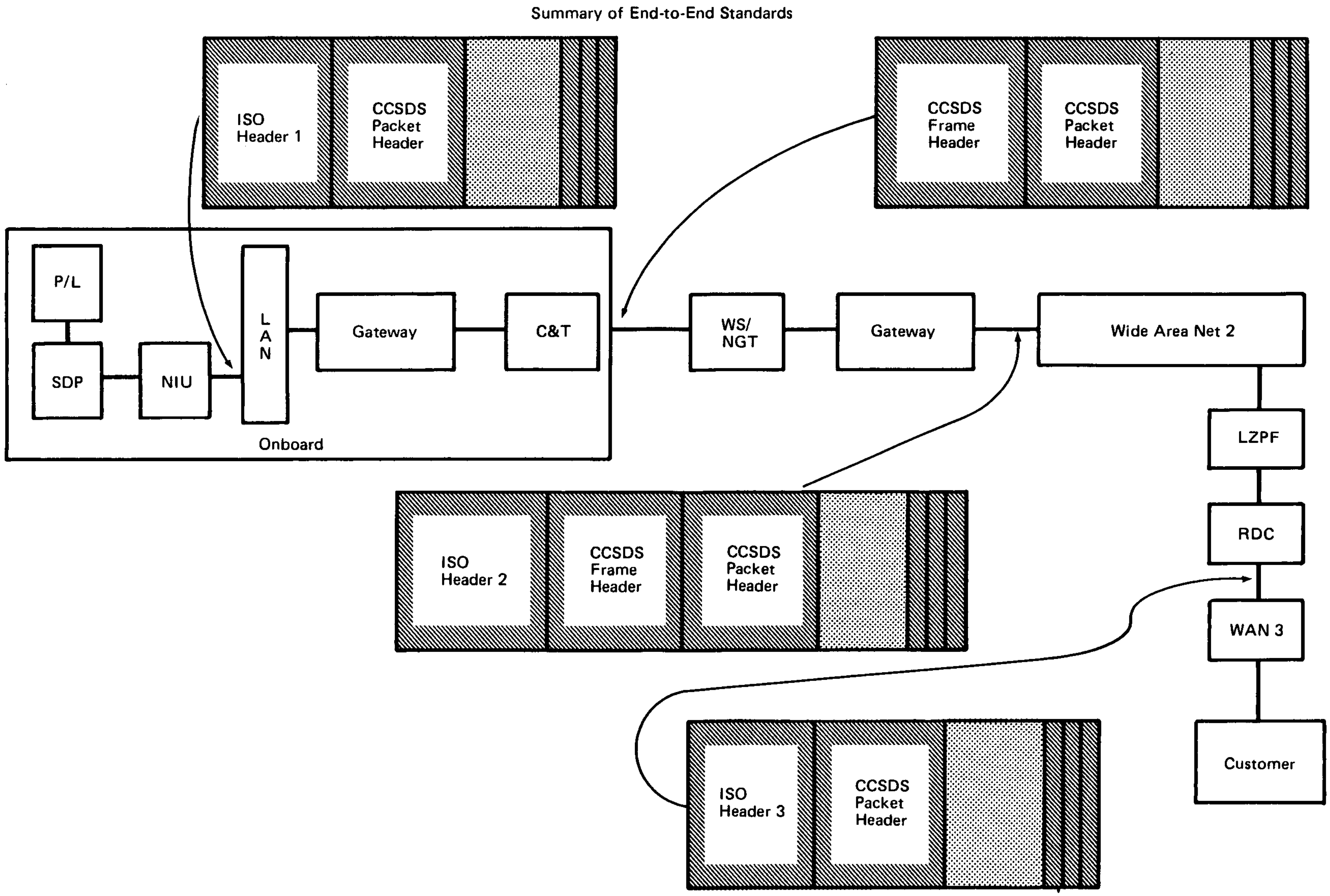


Figure 2.5.2-1. Example of End-to-End Application of ISO Headers

2.5.2.2.6 PRIVACY AND SECURITY LEVEL

Section 2.3, Privacy and Security, discusses various SSDS options. Most of these options will have little or no bearing on the Wide Area Communications Network. One exception would occur if one or more customers desired separate communication links with this separation occurring as soon as possible upon receipt of data on the ground. If data security is required throughout the network, in particular at each node, then significant design requirements will be imposed on the network. At this writing, there has been no specific direction to indicate the extent or type of security protection that must be implemented in the network design.

2.5.2.2.7 PERFORMANCE ANALYSIS

Section 2.6, Performance Analysis, also discusses various SSDS options. Several of the options reported in this section will have considerable impact on the Wide Area Communication network. Among these are:

- the techniques and extent of fault detection and associated data collection,
- the extent of overhead associated with data collection,
- the degree of automation of fault detection, and
- real time versus off line network performance monitoring.

The overhead associated with fault detection may require that all loads be interconnected by a separate network. The traffic on this network would be dependent on the degree of automation; the more fully automated, the less the traffic.

2.5.2.2.8 OTHER CONTROL CENTERS

In the overview section of this paper, other nodes such as OMV Control Center, OTV Control Center, Payload Operational Control Center, Polar Orbiting Platform Control Center, Co-orbiting Platform Control Centers, and Space Station Operational Control Center were listed. The design of these control centers represents a variety of options some of which are:

- the location of these control centers;
- the responsibility of the Payload Operational Control Centers (POCC);
- the extent of data transfers between centers; and
- functional allocation between control centers.

Each control center will require some space station status information which normally resides at the SSOCC. Consequently, the location of each of these control centers as, for example, the OMV control center will determine the communication links are required between these two nodes. Inorder to support the real-time command requirements it may be necessary to provide multiple communication links between all control centers that have the ability to support real-time access to data bases containing engineering data related to the SSPE associated with that control center. Clearly there are many options associated with the design of this subnetwork. For example, does any customer have the authority to access any related engineering data base at any control center whether that customer is located at a regional data center or at that customers' own facilities, further, access to data bases from several users always involves an orderly access protocol, and in many cases, a priority structure. Furthermore, the Platform Payload Control Centers may be distributed within the NASA organization as for example having a dedicated medical platform controlled from JSC and a dedicated materials platform located at MSFC while a dedicated science and astrophysics control center is located at GSFC. In such a location and control plan, communication links would be necessary to support each set of experiments. In the overall Wide Area Communication Network, it is conceivable that customers would be connected to the network without going through a POCC or through an RDC. In the event of unusual events such as an emergency or sudden unexpected change of mode, the customers must be advised of any circumstances that might affect their payloads; this factor implies the need for some core engineering data to be available at the POCC's and possibly even at some RDC's. Again, the general responsibilities of the POCC's could definitely have an impact on the communication links and data types transferred through the network.

2.5.2.2.9 COMMAND MANAGEMENT

The command management options are closely related to the Operational Control Center options. The location and distribution of the control centers will effect the various command management options. Among options to be considered are:

- o Should command management have a central control point;
- o If centralized, where will this be located; and
- o Location of restriction and constraint checks.

Many, if not most, of the constraint commands will require companion engineering commands to adequately insure the space station status. In these cases, engineering commands must be generated from the SSOCC and the processing and checking for constraint commands must be coordinated with that facility. It would appear that the final check on any constraints (and some restricted) commands must take place either at the SSOCC or in the space station itself. These options have bearing on the Wide Area Communications network design as constraint checking will require an evaluation of available resources not only in the Space Station but at all ground facilities associated with a data flow resulting from these commands. Such constraint checks must be made up to initiation of the command(s) and, therefore, require a real time subnetwork.

2.5.2.2.10 SWITCHING OPTIONS

This network will be required to transport several data types: high speed and low speed data, voice, video and commands. As stated previously, this network will have many destinations and sources. Given the variety of data types and the complex topology, several switching or routing options exist. These include:

- scheduled routing,
- static directory routing,
- dynamic directory routing,
- distributed adaptive routing, and
- mixture of the above.

Scheduled routing refers to a system where all circuits are established in accordance with a schedule. Obvious problems with this type of routing over a large network with diverse data types result in unexpected failures of degradations occur that require rerouting. A closely related technique for routing uses a pre-defined table indicating data paths to various destinations. This technique is referred to as static directory routing. Such routing tables are updated by manual intervention only.

By contrast, directory routing may be accomplished by automatic updates determined by network conditions or data type.

Adaptive routing would be distributed to nodes where routing decisions are based on such factors as error conditions and queue length.

The Wide Area Communication Network may be composed of a combination of these routing techniques based possibly on data rate. Nevertheless, the choice of routing technique would poorly impact the network design.

Other options associated with this network design include the implementation of message of packet switching on portions or all parts of the network.

2.5.2.2.11 CUSTOMER INTERFACES

2.5.2.2.11.1 Programmatic

Customer interfaces include programmatic options. Examples of such options involve initial customer familiarization with the Space Station program contract negotiations, and customer accounting and billing. Customer familiarization includes many options: seminars, news releases, articles, mailing service, public address and articles by NASA personnel, as well as a public by accessible electronic query information data base. It is likely that all of these options will be used in the course of the Space Station Program. Each customer will negotiate a contract with NASA stipulating the services to be provided, the associated prices and a set of requirements on both parties required to fulfill this contract.

During the mission, such contracts may require renegotiation depending on the progress of the mission and other factors such as customer needs for additional services. There are many options for the structure of such contracts: fixed fee, fix fee dependent on the level and quality of service provided, reimbursable funds between government agencies, prices based on NASA costs such as the original STS contract arrangements, etc. Certainly the determination of which party pays for the link from a regional Data Center to the customer facilities and under what conditions represents a set of options. Costs, tracking and billing offer many options. Billing can be performed on a near real-time basis (as the services are being provided) or, optionally, the entire cost can be based on a mission service independent of the specific functions that occur during the course of the mission. The former approach will place the burden on NASA to develop more thorough cost accounting, although it allows greater flexibility to the customer in adjusting real time operations to a given budget.

2.5.2.2.11.2 Physical

Physical interfaces can be categorized as: physical type, i.e., radio frequency or hardwire, and service type, i.e., archives, data reception and command transmission, etc. Hardware physical interfaces typically will be either copper or fiberglass. Radio frequency interfaces could include DOMSAT links among Regional Data Centers, a central ground facility, directly from the Space Station or from a Space Station-to-geostationary satellite-to-customer link. Each of these would have different physical options: r.f. interfaces associated with carrier frequencies; modulation type; encoding; antenna type; etc. Hardwired interfaces will likely be standardized with few options.

Other options associated with customer interfaces relate to data base access. To assure customer privacy, selected customer data will require access control. On the other hand, some customers will prefer free interchange of data among interested parties. Consequently, there will be different levels of data base access and therefore associated options. Other options include the design of these data bases or in such factors as whether they are centralized or distributed. Further, will some customers have the use of common subroutines.

Options associated with service interfaces include access to different data bases (such as the long-term payload archives, which will be pre-negotiated between the customer and NASA), and access to the engineering data set, which may require a different physical connection depending on the data system design.

2.5.2.2.12 NETWORK MANAGEMENT

Network management refers to the programmatic management of the network. This management could be centralized at a single location in NASA or distributed among various NASA organizations with specific responsilities associated with network functions. Options associated with network management essentially represent the division of responsibility within the NASA organization.

2.5.3 Local Area Networks

2.5.3.1 Description

A local area network (LAN) is an information transport system for information transfer between devices located geographically close. LANs generally provide high-bandwidth communication over inexpensive transmission media at a cost which is very low compared to the costs associated with traditional data communication networks. Multiple LANs may be interconnected by gateways/bridges providing an interconnecting vehicle for a wide variety of communications devices within an establishment.

Local area networks basically consist of transmission media, network interface units (NIUs), and the Network Operating System (NOS). Options for the transmission medium, NIU and NOS are considered in more detail in sections 1.7.1.1, 1.7.1.2, and 2.1.3 respectively.

The characteristics of a local area network are transmission medium, transmission technique, topology and media access method. The transmission medium of a LAN is the element of the network which carries the physical signals between nodes or stations. Currently practical transmission media are twisted pair copper wire, coaxial cable and fiber optics, among others. The physical signal can be applied to the transmission medium in one of two ways. In baseband signaling, the signal is applied directly to the medium as a series of digital voltage pulses. The other transmission technique, known as broadband signaling, employs the use of modulation to take advantage of the wider bandwidths available at higher frequencies. The manner in which the transmission medium interconnects the nodes of the network is defined as the LAN topology. Basic topologies are star, bus, ring and mesh. The LAN's media access method is the technique by which the nodes gain the right to transmit information onto the medium. The medium access can be either controlled access, such as token-passing and polling, or demand access, such as Carrier Sense Multiple Access (CSMA).

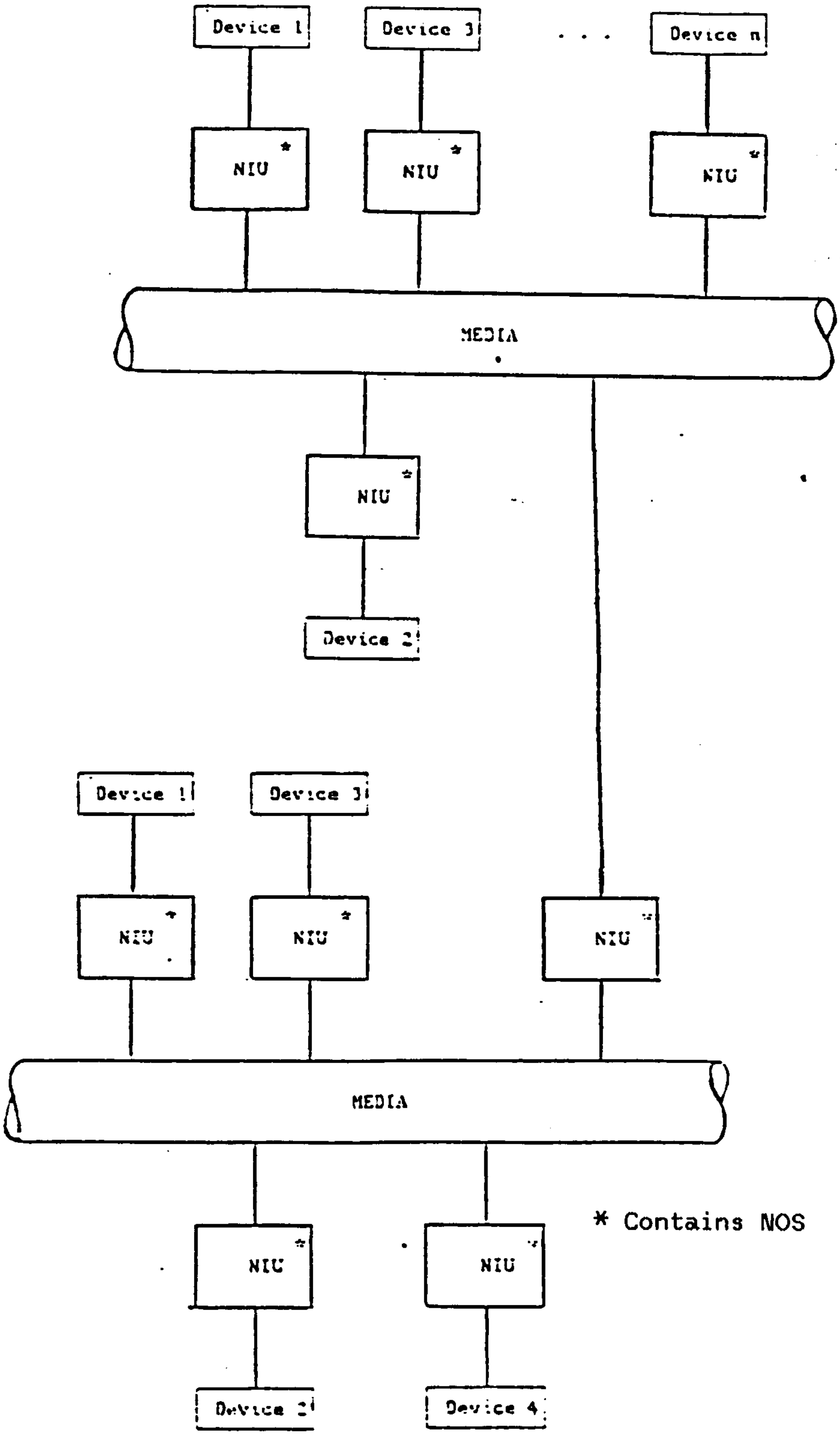


FIGURE 1. Generic Local Area Network

This paper addresses the types of technologies used to implement LAN's and compares various LAN implementations from a number of sources.

In general, the discussion on local area networks that follows applies to all orbital elements of the Space Station Program:

- The Space Station
- Co-orbiting Platform (COP)
- Polar Orbiting Platform (POP)

2.5.3.2 Option Characterization

The technological parameters that define a local area network are:

- Transmission Medium
- Transmission Technique
- Topology
- Media Access Method
- Network Interface Unit (NIU)

Each of these parameters are described below. They should be considered in terms of.

Cost. The cost of the network varies with the above parameters selected. The initial wiring, connection, maintenance, and reconfiguration costs should all be considered.

Growth/Technology Insertion Potential. The wiring distance and number of connectors varies depending on the topology selected. Some network configurations facilitate reconfiguration and growth while others do not.

Performance and Delay Characteristics. The access protocol has a great effect on network performance. The topology can also have a significant impact on the delay characteristics of the network.

Environmental Characteristics. The tolerance to radiation and suitability for implementation in space need to be considered.

Standardization/Commonality. Standards for some network configurations have been developed.

Physical Characteristics. The weight, volume and power required have a significant impact on the decision for an onbaord LAN.

Reliability. Single points of failure are more common for some topologies than others. Differences also exist in the reliability of hardware and software for a given topology and protocol.

Ease of Multiplexing. The multiplexing of voice, video and data on more than one channel can be done more easily and at a lower cost for some topologies and media than others.

Risk. The risk associated with the development and production of the LAN varies with the media topology, transmission method, media access method and NIU services provided. The LAN should be evaluated according to NASA's eight levels of technological readiness.

o Transmission Medium

Most LAN's now in use or being designed, use bit-serial transmission over coaxial cable or twisted-pair copper wire. However, the use of fiber optic media in LAN systems is increasing and will continue to do so in the coming years.

Unshielded twisted-pair copper wire medium is used extensively for normal analog voice communications. However, the impact of environmental noise is not as detrimental to analog voice signals as it is to normal digital LAN traffic. A higher-quality twisted-pair medium has a higher characteristic impedance and is shielded to reduce eletromagnetic interference, thus providing a reliable transmission medium for digital LAN systems operating at data rates between 1 to 10 Mbps.

Currently, coaxial cable is the most widely used medium for local area networking. Since coax is a shielded medium by definition, it is largely immune to electrical noise and carries data at higher rates over longer distances than can twisted pair. There are two general types of coaxial cable, named for the transmission technique that they support. Baseband coaxial cable carries one signal at a time, usually at rates from 1 to 10 Mbps, which can be time-division-multiplexed. Broadband coaxial cable can carry many signals at a time through frequency-division-multiplexing. Although data rates on any one frequency band (channel) are lower than those available with baseband coax (1 to 5 Mbps), the availability of up to 20 or 30 channels on a single cable greatly increases the amount of data that the medium can carry.

Fiber optic cable is the newest medium in the LAN market. Transmission of signals for a small number of Kilometers (but significantly longer than possible using twisted pair or coax) at bit rates up to 50 Mbps is feasible using fiber optics. Another advantage of optical fiber is that the effect of electromagnetic interference on the fiber is negligible. At the moment, however, it is the most expensive medium available, but as its use becomes more widespread in telephone applications, it should become less expensive.

Table 1 compares the transmission media available for local area networking. See 1.7.1.1 for more detailed information.

	Twisted pair wire	Baseband coaxial cable	Broadband coaxial cable	Fiber optic cable
Topologies supported	Ring, star, bus, tree	Bus, tree, ring	Bus, tree	Ring, star, tree
Maximum number of nodes per network	Generally, up to 1024	Generally, up to 1024	Up to about 25,000	Generally, up to 1024
Maximum geographical	3 kilometers	10 kilometers	50 kilometers	10 kilometers
Type of signal	Single-channel, unidirectional; analog or digital, depending on type of modulation used; half- or full-duplex	Single-channel, bidirectional, digital, half-duplex	Multi-channel, unidirectional, RF analog, half-duplex (full-duplex can be achieved by using two channels)	One single-channel, unidirectional, half-duplex, signal-encoded lightbeam per fiber; multiple fibers per cable; full-duplex can be achieved by using two fibers
Maximum bandwidth	Generally, up to 1M bps	Generally, up to 10M bps	Up to 400 MHz (aggregate total)	Up to 50M bps in 10 kilometer range; up to 1G bps in experimental tests
Major advantages	Low cost May be existing plant; no rewiring needed very easy to install	Low maintenance cost Simple to install and tap	Supports voice, data, and video applications simultaneously Better immunity to noise and interference than baseband More flexible topology (branching tree) Rugged, durable equipment; needs no conduit Tolerates 100% bandwidth loading Uses off-the-shelf industry-standard CATV components	Supports voice, data, and video applications simultaneously Immunity to noise, cross-talk, and electrical interference Very high bandwidth Highly secure Low signal loss Low weight/diameter; can be installed in small spaces Durable under adverse temperature, chemical, and radiation conditions
Major disadvantages	High error rates at higher speeds Limited bandwidth Low immunity to noise and crosstalk Difficult to maintain/troubleshoot Lacks physical ruggedness; requires conduits, trenches, or ducts	Lower noise immunity than broadband (can be improved by the use of filters, special cable, and other means) Bandwidth can carry only about 40% load to remain stable Limited distance and topology Conduit required for hostile environments Not highly secure	High maintenance cost More difficult to install and tap than baseband RF modems required at each user station; modems are expensive and limit the user device's transmission rate	Very high cost, but declining Requires skilled installation and maintenance personnel Experimental technology; limited commercial availability Taps not perfected Currently limited to point-to-point connections

TABLE 1. Comparison of Transmission Media, 1983 Data Pro Research Corp.

o Transmission Technique

The bits of information that are transferred within the LAN are applied to the transmission medium in one of two basic methods known as baseband and broadband signaling. A baseband LAN is one that uses digital signaling; digital signals are inserted on the medium as voltage pulses, without modulation. The entire frequency spectrum of the medium is used to form the signal, hence frequency-division-multiplexing cannot be implemented. Unlike analog signals, digital signals cannot easily be propagated through the splitters and joiners required for a tree topology (a variation of the bus topology).

Broadband implies the use of analog signaling. Frequency-division-multiplexing is possible: the frequency spectrum of the cable can be divided into channels or sections of the bandwidth. Separate channels can support data traffic, video, and radio signals. Broadband is inherently a unidirectional transmission (i.e., signals applied to the medium propagate only in one direction), since it is not feasible to build amplifiers that will pass signals of one frequency in both directions. Full connectivity is achieved by using pairs of channels designated for bidirectional communications. Broadband can be incorporated into both bus and tree topologies, in addition to other topologies discussed below.

o Topology

The topology of a network determines the manner in which the stations (nodes) of the network are interconnected. There are many ways to interconnect nodes depending on the communications requirements, reliability, medium, and redundancy of the network. The four basic topologies are the star, bus, ring and mesh (see Figure 2). Variations and combinations of these basic topologies can yield useful improvements in performance and should also be evaluated.

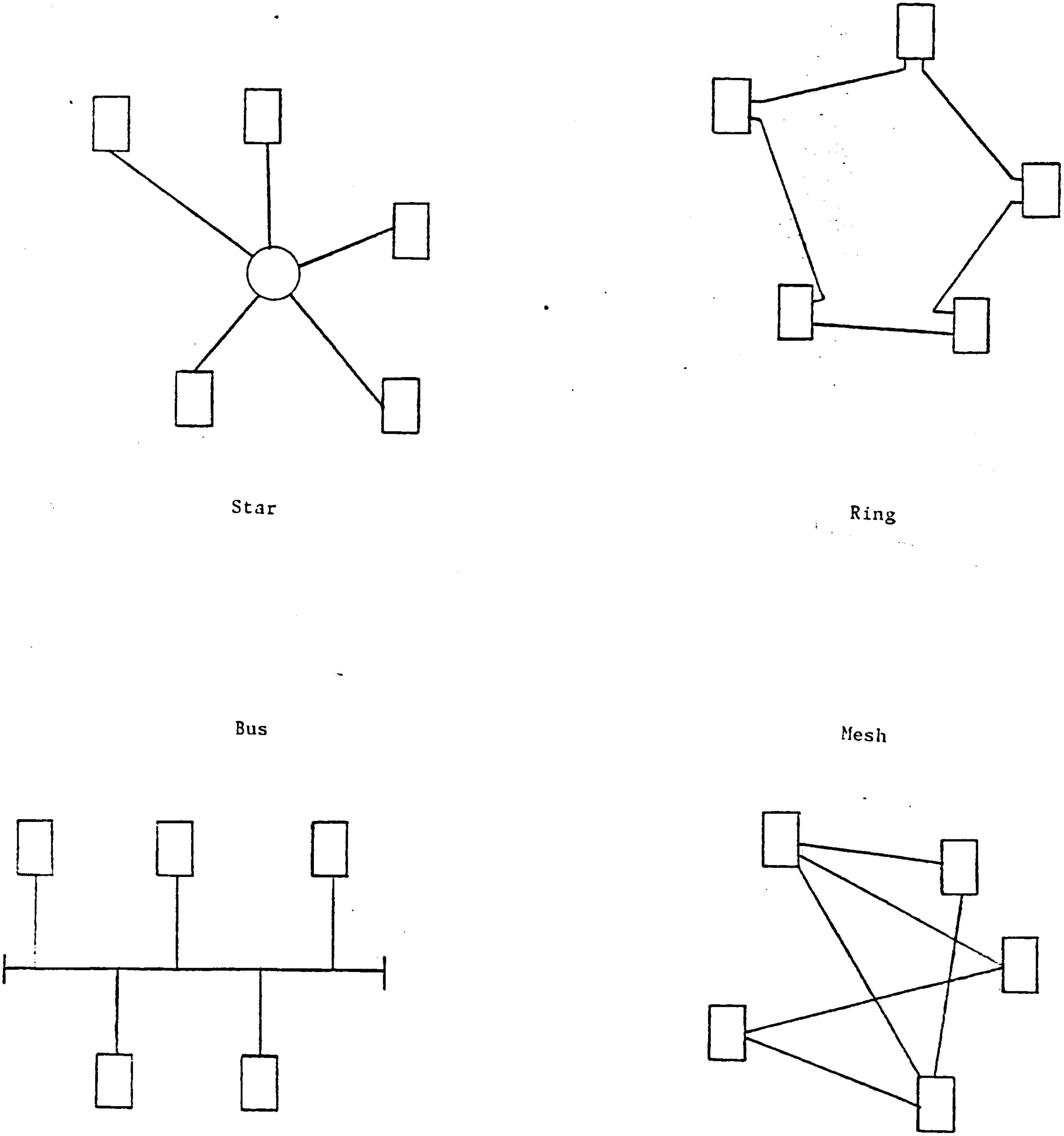


FIGURE 2. LAN Topologies

The Star Topology

In a star network, nodes are connected to a central star node through which all internode traffic must pass. The star node may be passive, as in many fiber optic stars, or active, as in most PBX's. A passive star splits the data signal, transmitting it equally to all nodes. An active star acts as a switch, setting up a direct connection between the transmitting node and the destination node.

Throughput is dependent on the network access protocol (discussed below). In an active star, the message processing time (spent at the star node) depends on the number of nodes, message length, processor speed, and message traffic statistics. The throughput rate for a given message is the reciprocal of the sum of processing time and 1/O buffering times.

Typical media access methods for the star topology are polling and token passing. (Media access methods are described in detail in section 2.5.3.2.4.) As long as the star node is capable of connecting to additional nodes, network changes are easily implemented. Wiring lengths, however, can be much longer than other topologies since each node needs a separate path to the star node.

Fiber optic passive stars are growing in importance, but they are usually limited to 16-64 nodes because the input power is divided equally among the other nodes. Due to marketing considerations, most active stars are limited to 1024 nodes. Most active stars, such as those in the PBX systems, use the twisted pair transmission medium. Because of the increasing need for PBX's, this technology is very developed, but support is primarily for relatively low data rates up to about 56 Kbps per circuit, with many circuits supported at the same time.

Fiber optic stars have been developed that work at approximately the same rate as point to point links (i.e., 50 Mbps and higher). Passive stars have the advantage of reliability. They are usually built as a fused biconical taper, which is simply a piece of fused glass with no electronic or moving parts.

Star networks have the general drawback of relying on the star node for all communications. Failure of this node brings down the entire network. On the other hand, the outlying nodes can fail with minimal impact to the rest of the network.

The Ring Topology

The basic ring topology is logically circular consisting of nodes connected to only two other nodes (Figure 2). The control of the network can be centralized in one node or distributed to all nodes. Data is transmitted unidirectionally around the ring. There are many possible medium access methods for the ring topology. The most common are token access, slotted-ring, register insertion, and polling (described in detail in section 2.5.3.2.4). Depending upon the protocol used, the message may be removed by the destination node or by the source node once it has travelled completely around the ring.

The performance of ring networks depends on the message transmission mechanism at the node. Some require an entire message address field to be buffered before retransmitting the message, which could cause long delays under heavy traffic. Others just examine one bit at a time and are able to send the message along with only one bit of delay. Once a message enters the ring, transmission time is usually not affected by traffic load. Where the number of nodes is relatively small and the distances travelled are short, the ring topology offers the best performance in high data rate applications (Reference 3). Rates of over 10 Mbps/s have been achieved with the token ring.

The most obvious problem with a ring topology is that failure of any node can potentially bring the whole network down. This problem has been solved, however, by various techniques such as device self-testing and self-elimination from the network. The elements of a node that could affect the network can also be made redundant and self-shorting.

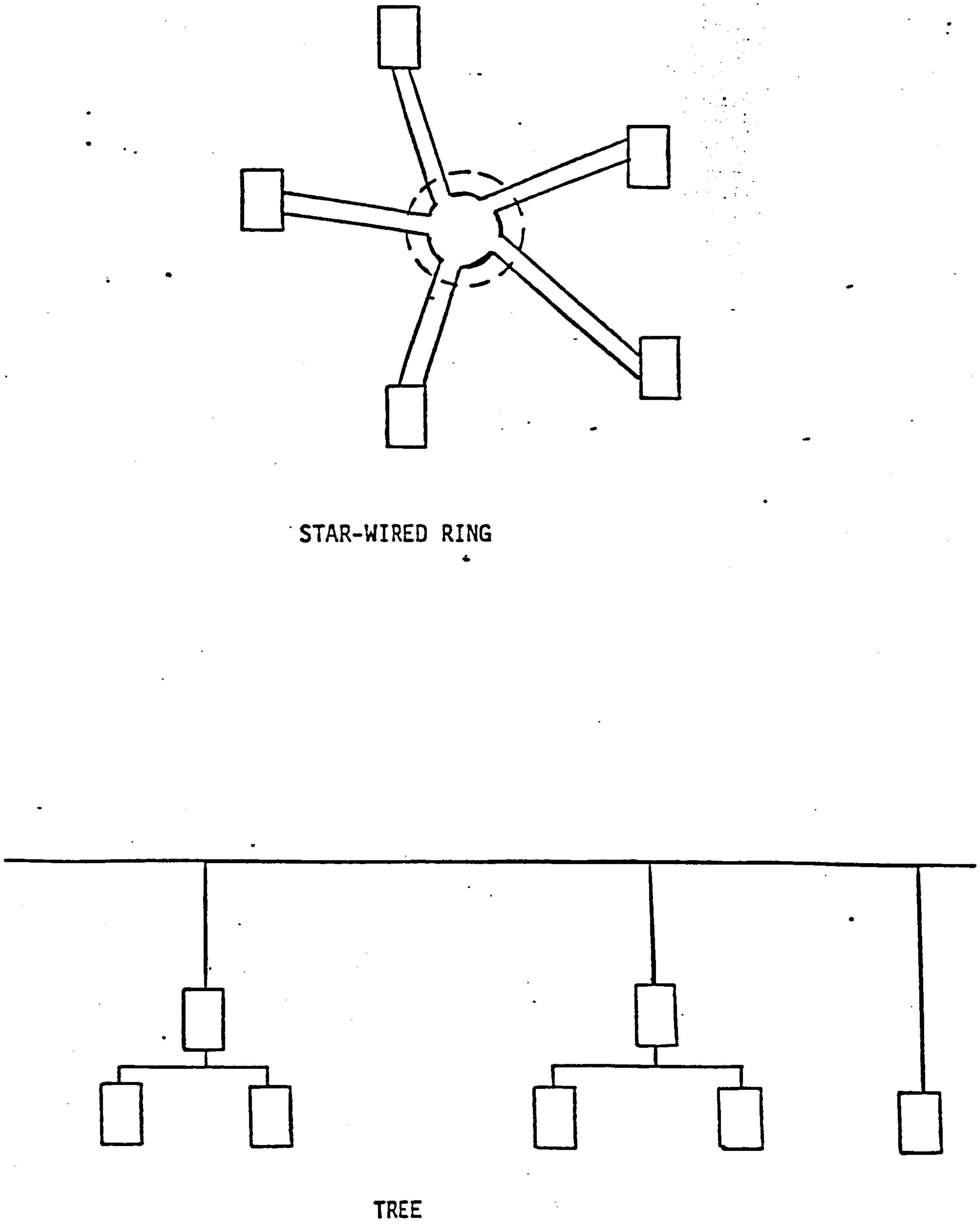
Note also that a duality exists between the ring and the star or bus. In the ring, if a transmitter fails "off", the node would not be able to regenerate the data and the entire ring would fail. In the star or bus with Carrier Sense Multiple Access with Collision Detection (CSMA/CD) if a transmitter fails "on", so that it always transmit data to the network, catastrophic failure also results.

Multiplexing different types of data on the ring is rather difficult because of the type of interface required, but it can be done. However, multiplexing analog video or voice could be uneconomical on a ring compared to other topologies.

Rings with various access protocols have been built in laboratories, and several have become successful commercial products.

The Star-Wired-Ring

The advantages of the star and ring topologies have been combined in a hybrid topology, the star-wired ring. This topology is physically similar to the star, but operates as a ring with unidirectional, point to point data transmission. (See Figure 3) Each node is attached to the wiring concentrator via a network interface unit. The wiring concentrator connects the nodes in a ring configuration, but maintains the capability of isolating an inactive or faulty station from the ring, thereby providing the fault isolation which the basic ring topology lacks.



STAR-WIRED RING

TREE

FIGURE 3. LAN Topology Variations

As with the star topology, wiring lengths can be much longer than with other topologies. Network changes can be easily implemented with this topology as long as the connection capacity of the wiring concentrator is not exceeded.

This configuration with the token passing media access method is the subject of the IEEE 802.5 standard and research at IBM.

The Bus Topology

In the bus topology, all the nodes connect to a single transmission medium. Frequency and time division multiplexing can be used with a bus topology. There are a variety of media access methods applicable to the bus including polling, token access, and CSMA/CD. Control can be centralized or distributed. A tree is an extended bus in which only one transmission path exists between two nodes (see Figure 3).

Bus performance depends on bus bandwidth, access protocol, number of nodes and the user traffic. The access protocol has the greatest effect on bandwidth utilization, with increased utilization usually requiring more complex protocols. CSMA/CD, for example, is a very simple protocol, but does not allow for very heavy bus utilization (Reference 3).

Typical data rates achieved by the bus are up to 50 Mbps over distances of 1 km.

One of the biggest advantages of the bus is that it easily allows for adding for eliminating nodes. Wiring is also minimized, but implementation with optical fiber is difficult. Coaxial cable is used most often because it is easy to tap inexpensively and with low losses. Several bus schemes have been implemented in commercial products. Most transient errors in a bus have the same effect as a collision, so the network can be very stable because a garbled message will just be retransmitted without long-term degradation of the network.

The Mesh Topology

In the mesh topology, there is more than one transmission path between any two nodes. The more "connected" the mesh is, the higher the number of redundant paths between nodes. The obvious advantages of the mesh topology is its ability to withstand a link or node loss without downfall of the network and to carry more traffic since more than one link can be carrying information at a time. The disadvantage of the mesh topology is the complex routing decisions which nodes have to make. Wiring complexity also grows more rapidly for mesh topologies than for other topologies. Adding a new node to the network involves connecting it to at least two other nodes, but if the additional node substantially increases the traffic, additional links may be required in parallel with those that are overburdened. The performance of the mesh depends on the exact topology and capacity of the links and on the routing algorithm used at the nodes.

Because of the routing computation complexity at the nodes, few local area networks are built around the mesh topology, but parts of LANs can use the mesh topology in order to increase redundancy in areas where high availability is required. Sensors, for example, can be connected to the processing elements of a LAN through a mesh network because they may be connected across long distances or through precarious regions. The mesh could then provide reliable and highly available interconnection.

In a totally connected mesh, each pair of nodes would be connected with a dedicated full-duplex point-to-point link. This is one of the most fault tolerant ways to connect computers, but the cost of wiring makes it prohibitive when more than a few nodes are required. It is also difficult to reconfigure.

o Media Access Method

The manner in which devices gain access to the medium is determined by the network protocol. The media access protocol and the network topology are interrelated; not every topology allows the use of every media access protocol.

Access to the medium may be either controlled or demand access. With demand access techniques such as CSMA/CD, a node attempts to gain access whenever it has a message to send. With controlled access such as token passing or polling a predetermined method is used to award access. The media access protocol options considered here are:

- Token Passing
- Slotted Ring
- Register Insertion
- Polling
- Laning Poll
- CSMA/CD

The performance, reliability, and complexity vary with each protocol.

Token Passing

The token passing control scheme is applicable to the star bus, and ring topologies. In token passing, unique bit patterns called tokens are passed from one node to another. A node which has a message to transmit waits until it receives the "free" token. Upon receiving the free token, the node may transmit the message. When its transmission is complete, the source node transmits a "free" token so that other nodes may access the medium.

Token passing offers the advantage of regulated traffic on the network and the option of message priorities, making it suitable for data, voice, and realtime applications (Reference 7). A disadvantage of the token ring is token maintenance. The token may be lost, or a "busy" token may erroneously circulate around the ring. One solution for this problem is the designation of one node as the token monitor. The monitor could issue a new token if it detects that the token was lost. It would also detect a continuously circulating "busy" token on a ring and correct this problem by setting the token to the "free" bit sequence.

Slotted Ring

This protocol is analogous to a passing train with empty and full cars. In the slotted ring, fixed length data slots continuously propagate around the ring. A bit in each slot indicates whether the slot is full or empty. A node with a message to transmit waits until an empty slot arrives, sets it full, and inserts the message into the slot. Each node monitors the ring for data addressed to it. In some slotted rings, the destination node can set response bits indicating any errors and message reception. The source node removes the data and sets the indicator bit "empty" freeing the slot. This node is now free to transmit another message.

An advantage of the slotted ring is that the nodes have minimal interaction with the ring. The slotted ring, however, is wasteful of bandwidth in terms of overhead. A slotted ring system, the Cambridge Ring, was developed by Cambridge University, and is now being commercially developed.

Register Insertion

With the register insertion control scheme, each node inserts the messages to be transmitted into a shift register. When the ring becomes idle, the contents of the shift register are inserted onto the ring. The next node then accepts the data on the ring into its shift register (buffer). If the message is addressed to another node, it is reinserted onto the ring. The destination node either removes the message or copies it while retransmitting it. If the message is retransmitted, the source node removes the message.

The register insertion protocol allows a higher ring utilization than any of the other methods. Priorities may also be used with this protocol. A major disadvantage of register insertion, however, is the possibility of an error in the address field which could cause the packet to circulate indefinitely. An example of the register insertion protocol implementation is the IBM Series 1 Ring which operates at 2 Mbps.

Polling

In the polling access method, one node acts as master of the ring or bus. This master node initiates all data transfers by polling the nodes on the network. If a node has data to transmit, it may do so when it is polled by the master node. A lot of overhead is utilized by the polling process, especially if there are many inactive nodes on the network. Another disadvantage of this protocol is its inability to tolerate faults.

Laning Poll

The Laning Poll media access protocol, which is applicable only to the logical bus, has been specified by the Charles Stark Draper Laboratory in their Advanced Information Processing System (AIPS) proof of concept configuration. With the Laning Poll protocol, nodes compete for access to the medium through a polling process. A node with a message to transmit begins the competition by inserting a "1" onto the medium. Each node wishing to transmit a message inserts its priority onto the medium one bit at a time starting with the most significant bit. The transmission medium acts as a logical "OR". Nodes are eliminated from the competition when their priority bit is a "0" and the result of the "OR" is a "1". The bit by bit comparison of priorities continues until all nodes have been eliminated except the node with the highest priority. This node may then transmit its message.

The SubACS program, a distributed network for nuclear submarines, uses a similar protocol for accessing a redundant, high bandwidth fiber optic bus.

Carrier Sense Multiple Access with Collision Detection (CSMA/CD)

CSMA/CD includes the following concepts:

CS is carrier sense which allows a node to transmit only if the channel if free, that is, if no other nodes are transmitting at that time.

MA stands for multiple access; any node needing to transmit may do so. If no acknowledge is received within a certain time, the message is retransmitted.

CD is collision detection and allows a node to detect conflicting use of the channel by two nodes. Because of propagation delays, two nodes can sense that the channel is free and both can begin to transmit. When the collision is detected, each node stops transmitting, waits a random amount of time, and then tries to transmit again. This is the basic CSMA/CD protocol as implemented by Ethernet on a coaxial cable bus.

Several variations of CSMA/CD have been presented in the last few years in an attempt to minimize the drawbacks of this protocol. CSMA/CD provides excellent delay characteristics at loading levels below about 15%. If traffic density increases much more than this, most of the channel bandwidth is used in carrying colliding messages, and delays begin to increase far faster than the offered loading level. Another problem with CSMA/CD is that it does not allow certain messages, or nodes to have priority over others. This may be a problem for realtime environment applications where a guaranteed maximum delay is required.

The IEEE 802.3 standard describes the operation of a CSMA/CD bus.

On CSMA/CD/DR (Deterministic Retransmissions), different stations have different time windows over which their retransmissions can occur. These time windows are changed sufficiently often, so that all stations are given fair treatment over a long time span compared to the rate of change of the time windows. The effect is to significantly reduce collisions because very few stations retransmit on the same time window at a given time. This scheme eliminates the problem of low throughput at high loads.

To allow for prioritization of messages, a variation of the above protocol is needed. Instead of time windows being selected "fairly", they can be made to be application dependent. Certain high priority messages would be allowed to select shorter back-off times than other messages. This protocol, CSMA/CD-DP (dynamic priorities), is implemented on a chip set developed by Advanced Micro Systems.

Another variation of CSMA/CD, CSMA/CD/TS is (CSMA/CD with time slots) a more deterministic method of media access. The network operates as a CSMA/CD network until a collision occurs. Then, the network operates in a time slot mode. After each node has had the opportunity to send one message, the network switches back to the CSMA/CD mode of operation. This type of protocol is being implemented by Sperry in FODS (Fiber Optic Demonstration System).

CSMA has also been implemented with collision avoidance (CSMA/CA). With CSMA/CA, a synchronization packet is sent to all nodes. Any node wishing to transmit requests a transmission slot by placing a one bit in the synchronization packet. Each node has a predefined bit location in the packet. All nodes are notified by the packet which nodes wish to transmit. The nodes then transmit in the sequence indicated by the synchronization packet. (Ref 18)

o Network Interface Unit

The NIU enables user devices to be connected to a local area network by providing conversion of user data rate and communications protocols to that of the LAN system and vice versa. By functioning as a gateway, the NIU can provide interconnection of LAN systems that use different protocols, whereas by functioning as a bridge, the NIU can interconnect LAN systems with similar protocols.

The design of most LAN systems is based on the reference model for open systems interconnection proposed by the International Standards Organization (ISO), shown in Figure 4. The model is not a standard for building equipment but provides a framework that allows data communications standards to be developed in an orderly and comprehensive manner.

Where network services physically reside within a LAN varies greatly from one product to another. The placement of network intelligence can be generally classified into one of several levels:

- The end-user device performs all station functions and drives the transmission medium; the LAN provides Level 1 services only.

- The end-user device connects to a NIU that provides Level 1 and Level 2 services and a small amount of buffering.

- The NIU provides end-to-end Level 3 reliability services, plus increased provision for buffering.

- The NIU provides source-to-destination Level 4 services such as error-free virtual point-to-point channels that deliver messages in the order in which they are sent.

- The NIU provides Level 5 session management services, plus recovery from broken Level 4 connections.

- The NIU is a micro- or minicomputer that provides Level 6 interconnection functions.

7	APPLICATIONS LAYER	o Serves the end user application process
6	PRESENTATION LAYER	o Performs data transformations o Data translation o Formatting
5	SESSION LAYER	o Establishes connection, or session, between two presentation layers o Manages the dialog o Provides circuit fault recovery
4	TRANSPORT LAYER	o Segmentation of message o Flow control o Ensures acceptable error levels
3	NETWORK LAYER	o Arranges message into packets o Routing o Congestion control
2	DATA LINK LAYER	o Formats data into data frames o Processes acknowledgement frames o Error handling
1	PHYSICAL LAYER	o The mechanical, electrical, and procedural interfacing to the media

FIGURE 4. Open Systems Interconnection Reference Model

Whenever a function resides in the end-user device, it is considered to be outside of the LAN. The user is usually responsible for developing and maintaining this software, although some LAN suppliers have begun to provide high-level software for certain applications.

For a further discussion on NIU's, refer to section 1.7.1.2.

Options

The options for the Space Station onboard LAN are a subset of the various combinations of the parameter options described above.

Due to the nature of the star topology (active), as well as the slotted ring, polling and the register insertion media access methods, they are not considered as an option. These techniques do not meet the requirements of reliability/availability and performance of the Space Station LAN. The major disadvantages of each are reiterated below.

Active Star –	Not fault tolerant long wiring lengths
Slotted Ring –	high overhead
Polling –	high overhead low fault tolerance
Register Insertion –	purge mechanism

The onboard LAN options are:

Token Ring

- o ANSI X3T9.5 FDDI

Token Bus

- o IEEE 802.4

CSMA/CD Bus

- o FODS

Laning Poll Bus

- o SubACs
- o AIPS

Others

- o Langley Mesh
- o SAE/AE - 9B

Table 2 provides a brief description of these networks. The Langley Mesh and SAE/AE-9B networks are not fully defined and, therefore, not included in the table.

2.5.3.2.1 Token Ring

The media access method for the token ring is token passing. In a token ring unique patterns called tokens are passed from node to node around the ring. A node which has a message to transmit waits until it receives the "free" token, then changes the token to the "busy" bit pattern and transmits the message behind the "busy" token. Each node monitors the ring for data addressed to it. The destination node copies the data as it goes by and appends some bits to the end of the messages to indicate any errors and messages reception. When the message complete its trip around the ring, the source node removes the message from the ring and transmits a "free" token so that other stations may access the ring.

Table 2: Non-commercial NIU's

NIU/ MANUFACTURER	IOS/OSI LAYERS	ACCESS METHOD	NETWORK TOPOLOGY	LEVEL OF REDUNDANCY	DATA TRANS.	DATA TRANS.	DATA TRANS.	VIDEO/ VOICE INTERFACE	BACKEND INTER-FACES	POWER REQUIREMENTS	ENVIRONMENTAL SPECIFICATIONS
SubACS/ IBM	1-5	Priority arbitration	Multiple buses (passive star)	1-2 (NIU is simplex path is redundant)	Fiber Optic wire	Baseband	64 Mbps	None	Navy devices, processors	Approx. 300W	Operating temp.: MIL-STD Rel. humidity: MIL-STD
AIPS/ CSD Laboratory	1-3	Laning Poll	Mesh	1-3 (NIU is simplex network is redundant)		Baseband	3 Mbps (later to 100)	Planned			Operating temp.: MIL-STD Rel. humidity:
FODS Sperry	1-2 increasing	CSMA/ CD/TS	Bus (passive star)	1-2 (NIU is simplex network is redundant)	Optical Fiber	Baseband	100 Mbps (later to 300)	Possibly	RS232 & a DMA high speed interface	35W (later decreased by 1/3)	Operating temp.: Rel. humidity:
ANSI x3T9.5 FDDI	1-2	Token passing	Ring	Counter rotating rings	Optical Fiber	Baseband	100 Mbps	Possibly			Operating temp.: Rel. humidity:
IEEE 802.4 Token Bus	1-2	Token passing	Bus			Broadband or Baseband	1 Mbps - 100 Mbps typically 5Mbps	None			Operating temp: Rel. humidity:

o ANSI Token Ring

The ANSI X3T9.5 Fiber Distributed Data Interface (FDDI) local area network is a 100 Mbps token passing ring with 4B/5B coding and a maximum packet length of 4500 bytes.

Free tokens are inserted directly behind the message. Therefore, multiple messages may exist on the ring, but there may be only one free token on the ring at any given time. This increases efficiency if the packets are small relative to the ring latency.

Priorities are allowed as are synchronous and asynchronous messages. When the ring load is low, both asynchronous and synchronous messages can be transmitted. If the load is high, only synchronous messages can be transmitted (i.e., synchronous messages have a higher priority).

The standard provides counter rotating rings for redundancy. The ring will continue to operate with one failure, and may operate with two or more depending on where they occur.

2.5.3.2.2 Token Bus

The operation of the token bus is similar to that of the token ring. A unique bit pattern (token) is passed from node to node on the bus. A node may only transmit a message when it has possession of the token. When the transmission is complete, the node inserts a "free" token onto the bus. The major difference between the token bus and the token ring is that the nodes in the token bus must insert into the token field the address of the node which should receive the token next, the nodes thus forming a logical ring. Thus, each node must know the address of the nodes preceding and following it in logical sequence. This requires more overhead than the token ring in order to accommodate the address in the token field. The token bus is also not as efficient as the token ring under heavy loads, and under light loads has a longer delay (see Reference 3).

The token bus is the subject of the IEEE 802.4 standard.

2.5.3.2.3 CSMA/CD BUS

The CSMA/CD media access method does not appear to be suitable for Space Station application because of its non-deterministic nature. However, a variation of the standard CSMA/CD protocol, FODS, a CSMA/CD/TS LAN, compensates for this effect.

- Fiber Optic Distributed System (FODS)

The FODS LAN being developed by Sperry under the direction of NASA Goddard Space Flight Center uses CSMA/CD/TS (CSMA/CD with time slots) media access method. With this method, there are two modes of operation: the CSMA/CD mode and the time slot mode.

When in CSMA/CD mode, a node with a message to transmit listens to the bus traffic. When the bus becomes idle, the node begins its transmission. If a collision occurs, the transmissions are halted and the BIUs switch to the controlled access mode. In this mode, there is no contention for the bus; each BIU is assigned a unique time slot (according to its address) of fixed width to begin transmitting. At the start of transmission, the countdown through the time slots is suspended. At the end of the transmission the countdown is resumed. Each BIU has one opportunity to transmit (one time slot each) in a controlled access cycle. When one cycle is complete, the BIU's switch to the random access mode.

At the beginning of the random access mode, each BIU with a message to send waits a random delay before transmitting in order to avoid collisions from accumulated messages. The bus then operates as a CSMA/CD bus.

Presently, the NIU design encompasses the physical and data link layers (ISO/OSI). The network layer and perhaps higher layers may be incorporated in the NIU in the future. In the demonstration system, the NIUs dissipate approximately 35 watts. After technology improvements, the goal is 10 watts dissipation per NIU.

C-5

The network consists of variable length fiber optic links interconnected by a passive star coupler. A maximum of 32 nodes (some of which may be bridges) attach to each star coupler. The demonstration system will operate at 100 Mbps baseband. This will later be increased to 300 Mbps.

2.5.3.2.4 Laning Poll Bus

o   SubACs LAN

The Submarine Advanced Combat System (SubACs) utilized the Distributed System Data Bus (DSDB) communications network developed by IBM. This distributed system was designed to be highly reconfigurable and fault tolerant. The probability of a successful packet transfer is 99%. Throughout the system, inter- and intra- cabinet, the same network protocol is implemented. The media may be either fiber optics or twisted shielded pair with no effect on the basic protocol of the bus. Fiber is used for the intra-cabinet buses, and wire is generally used within the cabinets. For example, the array bus and data processor bus (see Figure 5) are redundant 8 Mbyte fiber buses. Twenty port passive stars interconnect the BIU's/bridges.

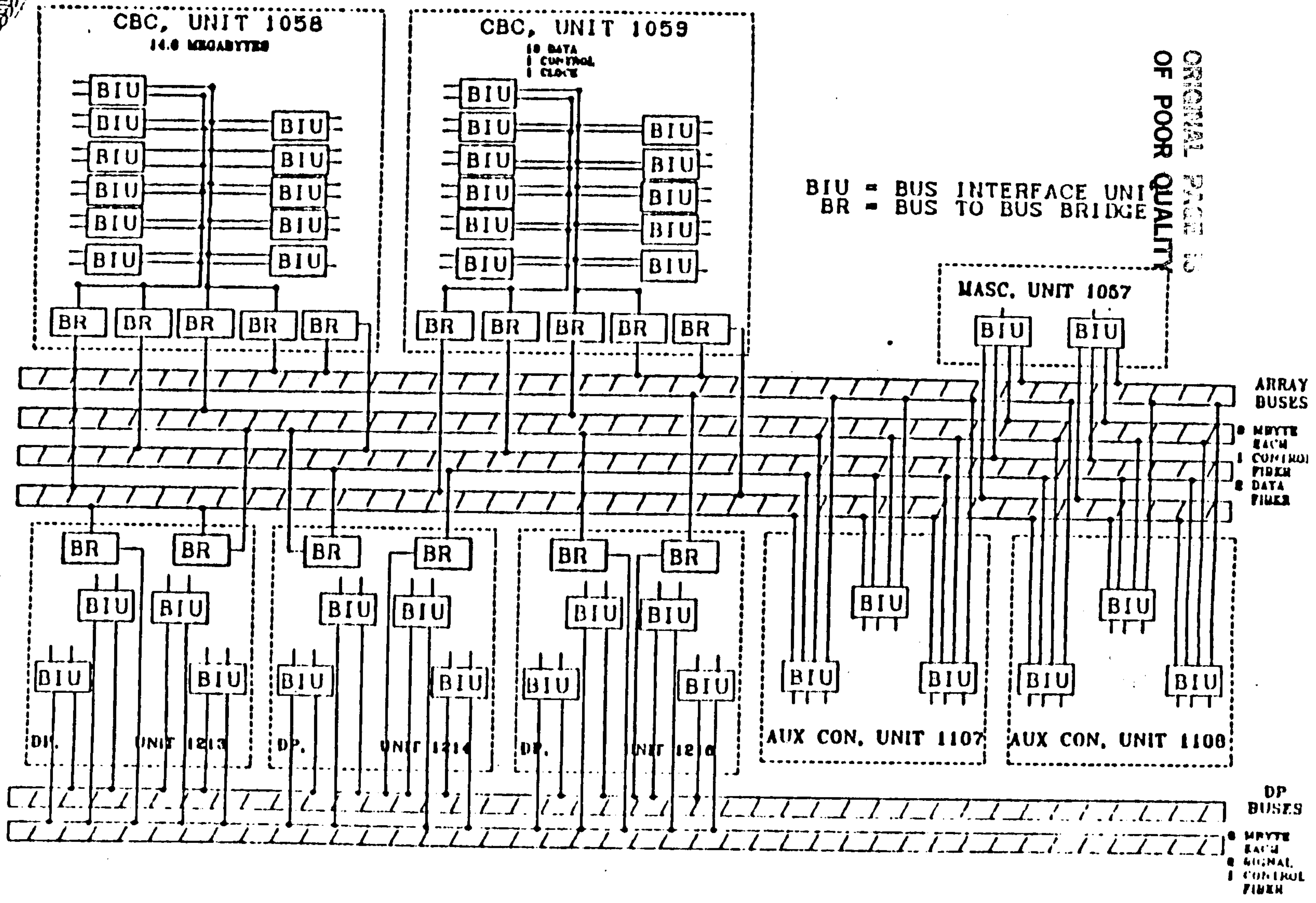
The DSDB handles both real-time periodic data and asynchronous data. The bandwidth is allocated to processes using dynamic rate monotonic priority scheduling in which periodic data has a higher priority than asynchronous data. DSDB also features packet switching and concurrent message handling.

Access to the media is obtained through priority contention which is similar to the Laning poll method (AIPS by the Charles Stark Draper Laboratory). Devices which have a message to transmit insert the priority of their message simultaneously onto the medium one bit at a time starting with the most significant bit. The priorities are logically 'OR'ed by the fiber optic medium. If a device which transmitted a 'O' priority bit received a '@' back, it is out of the contention. The bit by bit comparison continues until all devices have been eliminated except the device with the highest priority. In case of a tie between two messages with the same priority, the address of the device is appended to the priority for the bit by bit comparison.

ORIGINAL PAGE IS
OF POOR QUALITY

SUMMARY:

- o BIUs: 42 (22 WIRE, 20 FIBER)
- o BRIDGES: 16 (8 WIRE TO FIBER, 8 FIBER TO FIBER)
- o WIRE BUS UNITS - 2
- o FIBER BUS UNITS-6
- o FIBER STAR BOXES-6(20X20)
- o CBC BUS NODES-16
- o ARRAY BUS NODES - 13 EACH
- o DP BUS NODES - 14 EACH



DSDD BLOCK DIAGRAM

Figure 5.

The SubACS bus interface unit (BIU) and network operating system (NOS) can be briefly characterized as a modular, intelligent unit for message distribution in a local area network. The BIU is the hardware component in which the NOS software executes. The BIU/NOS include functions through layer 5 of the ISO/OSI reference model.

The following discussion is for the BIU and NOS as they exist in late 1984. The network and its capabilities have planned upgrades that will affect some of the details, but not the overall concept of operation.

The primary purpose of the BIU is to interface any of several processors or devices to the distributed system data bus (DSDB) so that those components can communicate in performing their functions. The BIU has a modular design both at the interface to the transmission medium and at the interface to a processor or device. A core set of modules and functions is used for all BIUs. The BIU can be personalized to connect to any of several Navy standard processor/device interfaces by the inclusion of the appropriate standard card and NOS software. A different medium or data transfer rate can be used by including the appropriate interface card, such as one to talk over a fiber optics link or another to talk over a wire line. SubACS does not include gateways, because the network was designed to have compatible protocols throughout they system and therefore needs no conversion functions supplied by a gateway.

A bridge is a special case of BIU which has been personalized with two media cards and the related NOS software in order to interface the transmission media of two networks. The bridge is used to selectively receive messages on one port for retransmission on the other port. SubACS includes both wire/fiber and fiber/fiber bridges between local area networks.

The NOS which executes in the BIU performs network failure management functions, in addition to normal message transmission and reception. At initiation of a connection between two ports. the NOS locates a path to establish as the virtual circuit, based on parameters of the new messages, and the failure status and loading of the parts of the network. At a failure, the NOS/BIU isolates the problem, reestablishes the virtual circuit, and continues operation automatically. Programs are moved to alternate machines at failure of a processor, if a suitable replacement is available. Failures are reported to the operator.

The BIU acts as a store and forward unit, first selectively receiving a message into a buffer in the BIU from the medium or attached device, then forwarding the message when the channel is available to its final destination. When functioning as a bridge, the NOS has the knowledge of which messages are to be ignored, and which are to be forwarded through to the other port after reception. The information about which processes and devices are active at each port is maintained dynamically as processes are activated or deactivated and as hardware is turned on or off or fails.

The BIU/NOS maintains a common time within the network by periodic equalization of local times to common reference. This time is used for various logs, allowing identification of the order of events across the network for problem analysis and evaluation of such performance parameters as time delays of messages.

The SubACs System uses standard software, wherever possible, and standard hardware. This provides for easy reconfigurability and simplified system integration. The redundant busses provide fault tolerance. Enhanced versions of the SubACs Architecture are planned for future development.

- AIPS

The Advanced Information Processing System (AIPS) designed by the Charles Stark Draper Laboratory (CSDL) is a fault tolerant processor complex which communicates using an Input/Output Network (I/O), an Intercomputer (IC) Network and a mass memory bus. Devices are connected to the networks via network interfaces. The I/O and IC interfaces are interconnected by switching devices (nodes) as depicted in the proof of concept configuration shown in Figure 6. The nodes and interfaces are described below.

The nodes are interconnected in a mesh configuration. However, the network operates logically as a bus. Under normal operations, the network configuration remains static. The advantage of the mesh interconnections is the ability to tolerate faults. When a fault occurs, the network is reconfigured to bypass the fault. The reconfiguration commands are sent to the nodes by the network manager.

Each node in the proof-of-concept configuration has five input/output ports. Three independent nodes provide redundancy. The node performs the following functions.

- Receives data
- Checks for protocol conformity and transmission errors
- Regenerate signal
- Transmits data
- Enables/disables port transmitters as commanded
- Circuit switching for fault recovery
- Responds to error and status requests

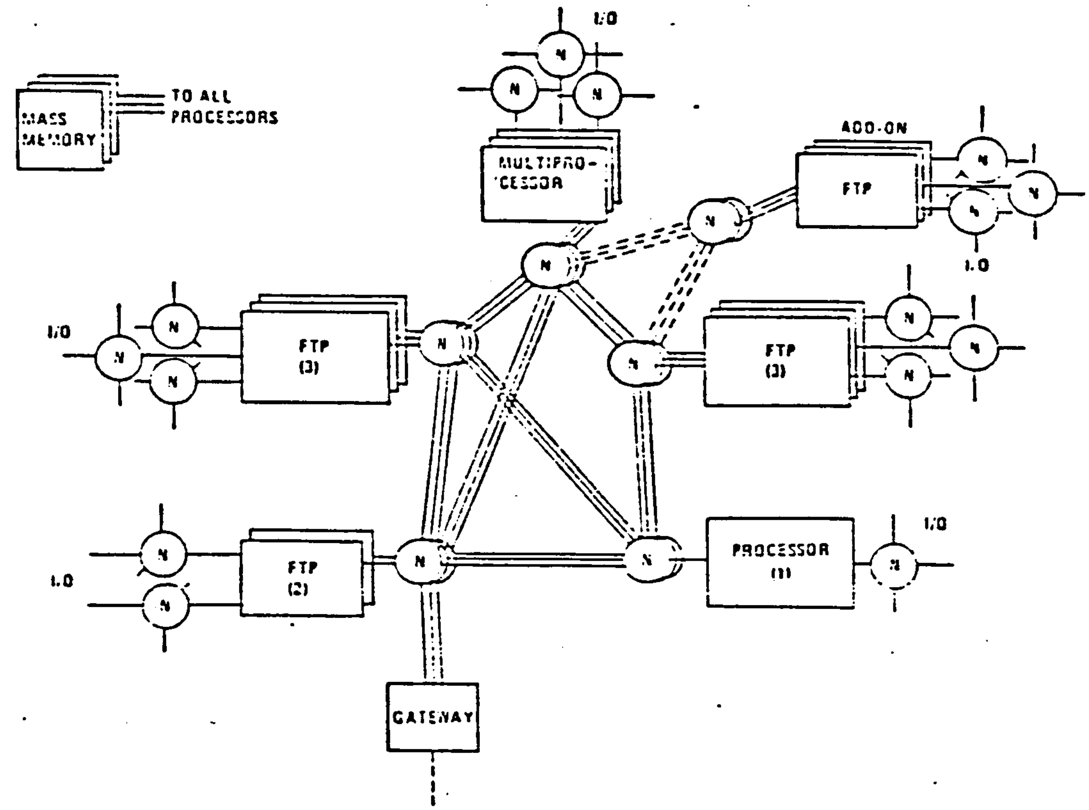


FIGURE 6 AIPS PROOF CONFIGURATION

The IC network is a cross-connected set of nodes which uses circuit switching for recovery from node-to-node links faults. Attached processors transmit on only one of the triply redundant IC buses, yet listen to all three. contention of the media is resolved using the Laning Poll technique. The IC network interface performs the following functions:

- Receives data
- Checks for protocol conformity and transmission errors
- Performs a bit by bit comparison of the three data signals from the triply redundant network
- Converts Serial/Parallel Data
- Decodes addresses
- Transmits data and enforces a transmission length limitation
- Enables/disables functioning of the interface
- Contends for the network using the Laning Poll technique

The I/O network is a cross-connected set of nodes which uses circuit switching for recovery from node-to-node link faults. It is a simplex network in which only one I/O processor is enabled to receive or transmit during an I/O exchange. Access to the media is gained through the Laning Poll technique. The I/O interface performs the following functions.

- Received data
- Checks for protocol conformity and transmission errors
- Converts Serial/Parallel
- Transmits data and enforces a transmission length limitation
- Enables/disables functioning of the interface
- Contends for the network using the Laning Poll technique

The AIPS also includes a triplex multiplex bus which provides communication between the mass memory and the general purpose computers (GPC). The interface to this bus, the mass memory bus interface, performs essentially the same functions as the IC network interface except address decoding, which is not necessary.

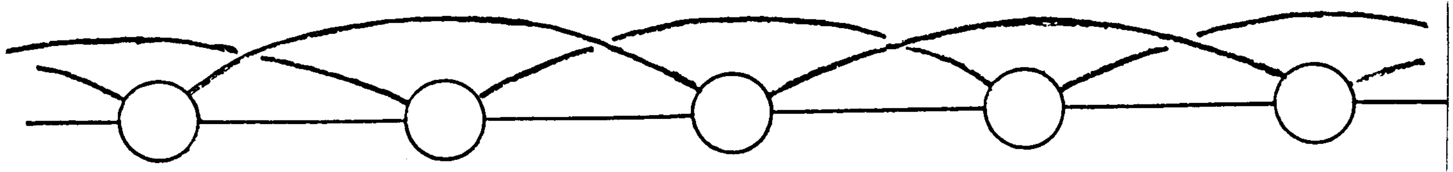
in order to perform these functions, the interfaces and nodes should contain the physical, data link, and the network layers of the ISO/OSI model.

2.5.3.2.5 Others

There are other systems which should be further investigated for possible application to the Space Station Program. these include SAE/AE-98 LAN and the Langley Mesh.

- Langley Mesh

The mesh topology currently being investigated at the Langley research center has been selected as a possible candidate to fulfill the reliability and fault-tolerance requirements of future space systems such as Space Station. Each node is connected to two adjacent nodes and to two more nodes further up and further back as shown in the figure below.



In other versions, even more connectivity is provided by adding links to the n+3 and n-3 nodes. The network could be laid in a linear fashion or looped back as a ring.

Currently investigations are under way to determine an access protocol and routing algorithm that would yield high performance. The present Langley emulator includes 7 nodes, each comprised of six 68000 processors for link control, node control, and interface management. Routing algorithm selection must be done with care because the overhead may reduce performance significantly. The purpose of the hardware emulator is to investigate among other things, the best routing algorithms.

Some of the problems with this topology for Space Station are:

1) Very early research stage so exact performance cannot be determined

2) Not a standard

3) Would be relatively higher cost because of more expensive node electronics

4) Does not necessarily meet FO/FS/R requirements, for example, when two adjacent nodes fail

An advantage is:

Should be more fault-tolerant than most other topologies considered

Another consideration is that performance will be highly dependent on the routing algorithm chosen

As the Langley research group further defines this network some of its disadvantages may be overcome, so their efforts will continue to be monitored.

o SAE/AE-9B The Society of Automotive Engineers (SAE) is currently working on a standard for a high speed local area network. The standard, SAE/AE-9B, will provide a high degree of fault tolerance and compatibility with military requirements. The standard will focus primarily on the physical and data link layers (ISO/OSI) and may be applicable to the network layer.

Some of the requirements for the SAE/AE-9B LAN are as follows.

- The Benchmark Information Rate
  - If 64 nodes each have 16 word messages to send, all nodes shall move their messages in less than 1200 usec.
  - If 16 nodes (out of 128) have 16 word messages to send, they shall all be sent in less than 300 usec.
- It shall support serial data transfer.
- It shall handle a periodic as well as periodic data transfers and support priorities.
- It shall support up to 128 nodes and a nodal separation of at least 300 meters.
- It shall provide for distributed control and possibly have a centralized mode available.

The topology and media access method have not yet been determined.

Characterization of Commercial LAN's

The Table that follows list a number of LAN alternatives from various commercial manufacturers. Following is an explanation of the entries used in the comparison matrices.

Topology: Possible network topologies (as previously discussed) are star, bus, ring and mesh.

Media Access Method: Current IEEE protocol standards are CSMA/CD, Token-Bus and Token Ring.

TABLE 2. CHARACTERIZATION OF COMMERCIAL LANs

LAN/MANUFACTURER	TOPOLOGY	ACCESS METHOD	TRANSMISSION TECHNIQUE	TRANSMISSION MEDIUM	PRINCIPLE HARDWARE COMPONENTS	OPERATING SYSTEM	DATA ENCRYPTION	END-TO-END ERROR RATE	REDUNDANCY
XLAN/Complexx Systems	Bus	CSMA/CD	Baseband	2-pair shielded wire	-Network Interface Unit (NIU)		None		
Net One/ Ungermann-Bass	Bus/Tree	CSMA/CD	Baseband/ Broadband	Ethernet Coaxial Cable/ CATV Coaxial Cable/ Fiber Optic	-NIU -Bridge -Gateway -Repeater -Headend	NOS		10^{-8} errors/bit	-Headend switch provides fault detection and automatic switchover to backup headend
Token Net/ Concord Data Systems	Tree	Token-passing	Broadband	CATV Coaxial Cable	-NIU -Headend	NOS		10^{-8} errors/bit	-Headend redundancy options available
Net Plus/InterLan	Bus	CSMA/CD	Baseband	Ethernet Coaxial Cable	-Transmitter-Receiver -NIU		None		
Local Net/Sytek	Tree	CSMA/CD	Broadband	CATV Coaxial Cable	-NIU -Bridge -Headend -Network Control Center (NCC)	NOS	Optional DES Algorithm	10^{-12} errors/bit	-Headend switch provides automatic switching from an on-line headend to a backup when parameters exceed specified fault levels -Multiple bridges may be cascaded to provide any level of redundancy
Contel Net/Contel Information System	Bus	CSMA/CD	Broadband	CATV Coaxial Cable/ Fiber Optic	-NIU -Bridge -Gateway -NCC	NOS			-Multiple NCCs may be installed on a network for redundancy
Ring Net/Prime Computer	Ring	Token-passing	Baseband	Biaxial cable	-NIU				

Transmission Technique: Possible network transmission techniques are baseband and broadband. In baseband transmission, the transmission medium carries one signal (always digital) at a time. In broadband transmission, the medium carries many signals (always analog) at a time, with each signal occupying a different frequency band on the medium.

Transmission Medium. This entry lists the possible transmission medium used for the LAN's principle data path (channel). Possible media are baseband (Ethernet) coaxial cable, broadband (CATV) cable and fiber optic cable. Principle hardware components: This entry lists those hardware components supplied by the manufacturer essential to the LAN. Possible components are:

- Network Interface Unit: Provides media access and user interface functions.

- Bridge: NIU that provides interconnection of multiple LAN's using identical protocols.

- Gateway: NIU that provides interconnection of multiple LAN's using dissimilar protocols.

- Headend: In broadband systems, signals are inherently unidirectional (i.e., they propagate in only one direction). Therefore, two data paths (channels) are needed to achieve full connectivity. The channels are joined at a point on the network known as the headend. All NIU's transmit on one channel toward the headend (inbound or reverse direction). Signals received at the headend are then propagated (and possibly amplified and/or frequency shifted) along a second channel away from the headend (outbound or forward direction). All stations transmit on the reverse-direction channel and receive on the forward-direction channel (see Figure 5).

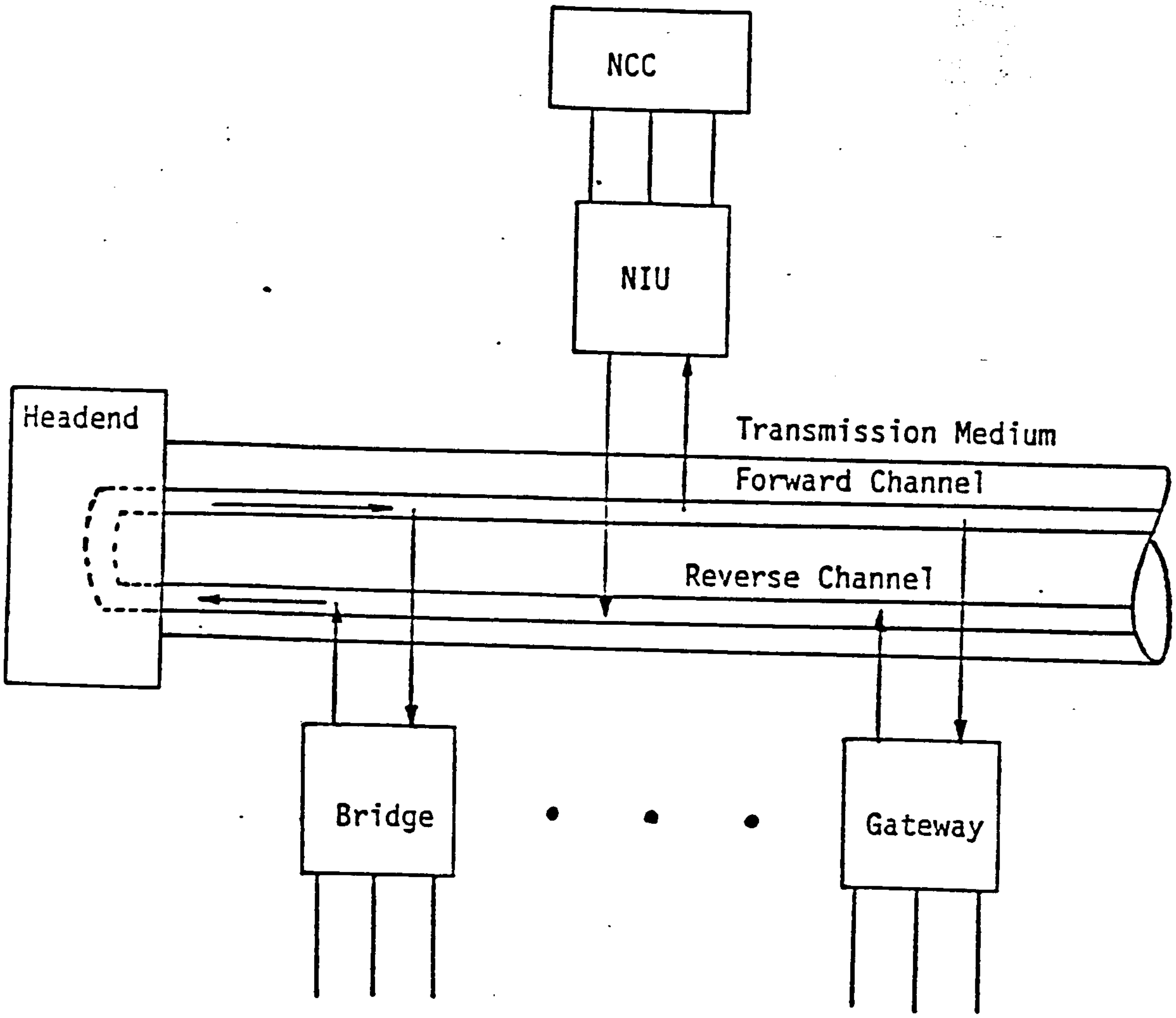


FIGURE 5. LAN Hardware Components

- Network Control Center. This device usually attaches to the network through an NIU and consists of a keyboard/screen interface and a microcomputer. It supports key operations, administration, and maintenance functions. All NCC functions can fall into one of three categories. configuration functions, monitoring functions, and fault isolation.

Operating System. Operating systems for networks are generally classified as either Network Operating Systems or as Distributed Operating Systems. In NOS, each host connected to the LAN runs its own (non-network) operating system. The networking is controlled by user programs that run on the various hosts. In DOS, the individual host operating systems are discarded and a single homogeneous operating systems is implemented for the entire network.

Data Encryption: Encryption can prevent unauthorized utilization and disclosure of network resources. A number of schemes for encryption exist, one of them being the National Bureau of Standards Data Encryption Standard (DES). The data encryption can be end-to-end or link oriented. End-to-end encryption is handled as a presentation layer function. In link-level encryption, all data plus all the headers except the layer 2 header are encrypted at the data link layer.

End-to-End Error Rate: This is the number of erroneous bits transmitted between end users.

Redundancy: The additional number of extra LAN components (resources) to provide tolerance of faults. If a component fails, then an equivalent component picks up the slack. Some systems use an approach in which the identical components function in parallel. Other systems activate the redundant components only upon a failure.

2.5.3.3 Projected Capabilities

TBS

2.5.3.4 References

1. Stallings, William. "Local Networks". ACM Computing Surveys, Vol. 16, No. 1, March 1984

2. Strole, Norman C. "A Local Communications Network Based on Interconnected Token-Access Rings: A Tutorial. "IBM Journal of Research and Development, Vol 27, No. 5, Sept. 1983

3. Bus, Werener. "Local Area Subnetworks: A Performance Comparison." IEEE Transaction on Communications, Vol. 29, No 10, Oct. 1981

4. Parker, Richard & Sydney F. Shapiro. "Untangling Local Area Networks." Computer Design, March 1983

5. "An Introduction to Local Area Networks." IBM, Second Edition, July 1984

6. Dixon, R. C.; Strole, N. C.; Markov, J. D. "A Token-Ring Network for Local Data Communications." IBM Systems Journal, Vol. 22, No. 1-2, 1983

7. "Local Area Networks: A Review." IBM, First Edition, Sept. 1983

8. Tanenbaum, A.S. "Computer Networks." Prentice-Hall, 1981

9. Clark David D. "An Introduction to Local Area Networks." Proceedings of the IEEE, Vol. 66, No. 11, Nov. 1978

10. "Networks and Architectures." 1983 Data Pro Research Corp.

11. "Hart Requirements for the SAE/AE-9B High Speed Data Bus." SAE/AE-9B HSDB Subcommittee, Issue #1, December 14, 1983.

12. Kossler, Paul A. and Robert L. McCaig. "SubACS Brings Distributed Processing to the Submarine Combat system." IBM Technical Directions, Vol. 9, No. 2, 1983.

14. "AIPS System Specification," The Charles Stark Draper Laboratory, Inc., Cambridge, Mass. NASA-JSC NAS9-16023.

15. "FDDI Token Ring Physical Layer Protocol Standard," Draft Proposed American National Standard, Nov. 30, 1984. X3T9/85 - X3T9.5/83-15 Rev. 8.

16. "FDDI Token Ring Media Access Control," Draft Proposed American National Standard, October 26, 1984. X3T9/84 - X3T9.5/83-16 Rev. - 7.2.

17. Murray, Nick. NASA Langley Research Center, Hampton, VA.

18. "MDC Local Area Networks Technology Study," McAuto, July 1983.

2.6 Network Performance Assessment

This section describes the options for monitoring the performance of the SSDS network including:

- Techniques for collecting and analyzing network activity on both a long-term and peak period basis.
- Use of specialized tools for testing trunk and gateway performance.
- Alarm mechanisms for reporting unusual events such as degraded user response, gateway failures, or poor trunk bit error rates (BER's).
- Monitoring of service quality for externally supplied communications links.
- Modeling tools for predicting network static and dynamic performance.

It is anticipated that the SSDS will use a combination of centralized and decentralized tools to assess network performance. Major requirements drivers in the SSDS network performance assessment subsystems are the needs for:

- Comprehensive reporting of performance information to a location where network activities can be coordinated, and network diagnostic information can be interpreted by expert personnel.
- Automated systems which facilitate the analysis of network traffic and performance statistics for long-term planning.
- The availability of modeling and simulation tools which allow the optimization of network performance with minimal waste of bandwidth and good user response characteristics.
- Data collection and fault detection mechanisms sufficiently detailed to allow immediate detection of system faults and early warning of incipient problems.
- Minimal use of manpower for routine activities such as monitoring gateway performance or link status.
- Quick reporting of network malfunctions with sufficient information to localize the nature of the problem.
- Minimal overhead in data collection or effect on network performance.

Major relevant questions associated with network performance monitoring options are:

- How centralized should the performance monitoring function be?

- Should a single facility be responsible for the monitoring of space segment LAN's, space-ground links, and the ground data distribution system?

- If not, how should these functions be partitioned and coordinated to maintain full cognizance of overall network performance?

- Should some sort of standardization be applied to network performance reporting? (e.g., hardware monitors or diagnostic interfaces)

- What types of tools should be developed to model and analyze data?

- What services for monitoring this data should be provided to network operators and users?

2.6.1 Hardware Configuration Options

The major set of hardware options in network testing and monitoring are those associated with the physical distribution of the hardware and its interconnection. Effectively this system should be considered a subnetwork and be configured according to its own specific requirements, with the same options associated with the design of a fault-tolerant network. Many current networks, including NASCOM, tend to deemphasize this superstructure resulting in significant manning requirements at network nodes. These manpower requirements can be significantly reduced through appropriate monitoring system design, and choices associated with such automation is an important set of options.

Since rapid location of network performance degradation and malfunctions is a major function of monitoring hardware, its organization is usually hierarchical in modern networks, with monitoring equipment located at each gateway feeding into progressively higher levels of network organization. Whether this organization should be used for the SSDS monitoring system; the relative depth of hierarchy, and the level of monitoring intelligence at each node are significant choices. In particular, the structure and the complexity of the SSDS control and diagnostic facilities are functions of these choices. The level of detail at which services are monitored is also a significant choice. Should network performance and quality be monitored only for major trunks and gateways? Is it feasible to provide comprehensive end-to-end monitoring of link quality for individual low-rate data link users (e.g., a 1200 baud voice-quality link)? At what level of network service detail should monitoring occur?

Use of a comprehensive hardware monitoring system throughout the SSDS implies the standardization of interfaces for performance monitoring and diagnostic equipment. Typical standards associated with such interfaces in modern communications system are RS-232C, IEEE 488, CAMAC, and STD. All have relatively low-level ISO physical layer descriptions with each vendor providing its own standards for requesting test sequences or performance data. Consistently applied high-level interfaces (e.g., formats for reporting trunk BER's or unusual events such as the failure or reinitialization of a gateway) do not seem to exist. Should NASA develop its own set of standards or should it cooperate with an organization such as ANSI or ISO to develop a standard?

Another option is the degree of intelligence for monitoring and diagnosis at each gateway. A trend in network design is the use of microprocessors at each network node to concentrate gateway statistics and perform local diagnostics. Locating logic and associated data buffering at each gateway has many advantages:

- Delays associated with diagnostic polling are minimized.

- Significantly more complex diagnostics can be integrated with routine system monitoring functions since only local bandwidth is used for diagnostic activities.

- More comprehensive historical information on intermittent system problems can be collected.

- Loopback testing can be done on a local basis, simplifying the location of malfunctions.

- Equipment can be configured for specific local subnetwork diagnostic requirements and problems.

- Local personnel can have access to diagnostic capabilities.

- Network control center structure is simplified.

- Additional functions such as break-in detection and thermal control are easily integrated.

Negative aspects are:

- The increased cost associated with providing duplicate monitoring and diagnostic equipment at many sites.

- Possible undetected node failure when diagnostic and gateway equipment simultaneously malfunction (e.g., power failure).

The means of interconnection of performance and diagnostic equipment has several associated options. The main concern is the possibility of simultaneous failure of network and diagnostic links. The options are:

- Use of standard network channels--Diagnostic information is communicated along the same channels as network traffic, either through subchannels or data headers. The main advantage of this approach is that is requires minimal resources. For systems with minimal alternative data channels (e.g., space-ground link) it may be the only feasible approach.

- o Use of parallel channels--Additional dedicated diagnostic lines are allocated in the network structure. An example of this mechanism are the diagnostic channels associated with AT&T Dataphone modem networks.

- o Use of physically separate channels--Diagnostic information is communicated in a manner which is fully separate from network channels. An example of this are satellite communications networks which use ground-based links for status polling and diagnosis.

2.6.2 Loopback Testing and Calibrated Signal Techniques

Most of the anticipated testing and performance monitoring associated with the SSDS is likely to use digital techniques. Such techniques may be based on the detection of faults through the monitoring of error detection/correction activities at gateways, i.e. through checks of polynomial encoding or simple block count checks. However, once problems are detected localization is generally done through loopback techniques, i.e. to test the performance of a remote line a loopback circuit is set up, a pseudo random pattern is transmitted, the received signal is compared, and associated BER's are determined.

Options for the SSDS in this area include:

- o Whether special digital loopback equipment designed to handle data rates higher than the current commercially available equipment should be developed.

- o To what extent the SSDS network should be designed to support loopback testing? How should the space segment be accommodated?

- o What sorts of associated analog equipment for measurements such as line loss, signal to noise ratio, modulation level, and waveform distortion measurements in high-bandwidth lines must be developed?

- o What level of automation should be associated with loopback testing? Should the testing be operator-mediated of fully automated?

2.6.3 Real-Time versus Offline

Monitoring of network performance may be done on a real-time or offline basis. Provision of real-time monitoring services is strongly affected by how data is collected. Types of real-time monitoring capabilities which may be of use in the SSDS include:

- o Trunk and gateway bit, block, and frame traffic rates

- o Trunk and gateway bit, block, and frame error rates

- o Node to node average and peak packet/message transmission times

- Retransmission rates
- Blocking rates for switched circuit traffic
- Packet and message processing rates
- Dynamic rerouting activities, particularly those related to gateway failure, overflow, or burst pattern changes
- Collision rates for individual local area networks
- System logins and sessions
- Power supply performance
- Intrusion and physical damage alarms

2.6.4 Software

Several types of services must be provided for SSDS network monitoring. We divide them into five classes. The allocation of functions to each set of services is a significant set of options. We do not consider Security Reporting in detail here, but it is likely to be integrated with other network monitoring services.

Local Reporting Services--Support collection of data for local use (e.g. internal LAN traffic) or for summary reporting to the network control center (e.g., session detail records).

Network Reporting Services--Support reporting of subnet, gateway, and trunk status throughout the network in a consistent manner.

Security Reporting Services--Support collection of data essential to monitoring network security, e.g., attempts to gain access to protected network resources.

Privileged Monitoring Services--Support monitoring of network performance by network operations personnel. Typically these services include monitoring activities which may affect network performance if utilized by a large class of network users or those which report information which might compromise network security.

User Monitoring Services--Support user monitoring of network performance allowing users to determine current network status, determine data link quality, and link alternatives when non-transparent services are used (i.e. when users are allowed to determine which network resources are utilized). Also services which allow users to register trouble reports.

2.6.5 Modeling and Simulation Techniques

Tools for predicting network performance fall into the following categories:

- o Analytic queuing models which provide numeric solutions based on probability distributions (e.g., Poisson, Erlang, etc.) and on estimates of line traffic rates.

- o General simulation tools such provide a framework for network performance prediction, but which are not limited to this activity.

- o Telecommunications simulation tools which typically include facilities for analysis of network specific structures such as adaptive routing, propagation delay, protocol performance, and tariffs.

Some common (and well-supported) tools associated with analytic and discrete simulation modeling of network performance are:

GPSS
SIMSCRIPT
SLAM
DSDS
SIMULA

Tools with specialized telecommunications modeling capabilities include systems such as the Telco Optymizer's or the DMW Teletraffic Optimizer. These tools include regularly updated tariff data bases, provisions for user-defined cost models, and simulation capabilities for the telecommunications-unique structures mentioned previously. However, since large voice-data-video networks are a relatively recent phenomena, most are geared toward the analysis of traffic on voice network. Although manufacturers are gearing new network analysis tools to the new network configurations, it seems unlikely that a vendor-supplied tool fully suitable for SSDS modeling will be available, and it is an option for NASA to consider the development of such a tool.

A major set of options associated with system modeling involves the means for introducing data into the model. Typically this has been a manual operation, and the automation of this process for the SSDS is seen as a means of reducing the labor associated with network analysis, and increasing the responsiveness (and usefulness) of the network model. The data collection and concentration activites are an appropriate consideration in the planning of the SSDS.

2.6.6 Performance Data Analysis

Analysis of network performance data serves the following purposes:

- o To facilitate system planning and resource allocation

- o To evaluate current system performance and user response

- To determine system reliability and availability

- To predict failure trends and target preventive maintenance

One option associated with performance analysis is the type of software tool to be used. Query systems associated with DBMS's can provide a partial solution, providing essential summary reports. More general analysis can be performed using a modern statistical analysis system such as SAS or SPSS-X. Such systems have standard interfaces with DBMS's and provide high-level tools for the generation of both production and ad hoc reports. Another option is to develop custom software for report production.

Types of analyses options which may be appropriate to the monitoring and prediction of SSDS performance include:

- All the reports mentioned in 2.6.3 with short- and long-term trend analyses

- Hourly utilization rates for gateways and trunks

- Analyses to locate dead circuits such as short and long call rates, and detailed circuit utilization reports

- Equipment failure models

- Trouble report summaries organized by location, equipment type, and user

- Data quality summaries

- Peak response and node-to-node propagation times and variances

2.6.7 Performance Degradation Alert and Diagnosis System

The major types of options associated with network performance degradations alerts are:

- Detection--Methods of detection must preclude the possibility of undetected gateway failure. Sufficient link redundancy must be provided to allow diagnosis in the presence of gateway failure.

- Filtering--Typical large-scale networks are subject to many spurious alarms which must be filtered so that serious problems are recognized through the noise. Examples of spurious alarms are those associated with equipment temporarily being taken offline for maintenance or those associated with routine network reconfiguration.

- Localization--Sufficient information must be communicated that the problem is understood and localized to appropriate equipment (or software). In a network involving many thousands of gateways, each with redundant subsystems, this may not be an easy task.

Techniques for detecting performance degradation or gateway failure include:

- Centralized polling--Gateways are regularly polled for status by a central control and status system.

- Centralized reporting--Gateways regularly send status messages to a central control and status system.

- Centralized deadman timer--If a gateway has not been active for a period of time then it is polled by a central control and status system.

- Paranoid democracy--Gateways expect regular "I am well" messages or handshakes from their neighbors. Otherwise they reroute traffic.

Spurious status messages may be filtered by operator-settable warning thresholds based on message type or frequency. Options include the degree of operator control, the types of thresholds used, and the level of machine intelligence involved (e.g., use of AI techniques).

Localization of failure or performance degradation presents another set of options. Presentation of diagnostic information may involve a simple alphanumeric description of the location and type of equipment involved or it may be a sophisticated graphics display with menu options allowing the operator to quickly recall a map of the equipment configuration with interconnections and affected subsystems clearly displayed. The second option is significantly more costly to develop, but saves time and manpower in long-term operations.