VALIDATED FAULT TOLERANT ARCHITECTURES FOR SPACE STATION

by

Jaynarayan H. Lala

Presented at

The Workshop on Technology

for

Space Station Evolution

Dallas--Fort Worth, Texas

January 17, 1990

The Charles Stark Draper Laboratory, Inc.

Cambridge, Massachusetts 02139

OUTLINE

- Fault Tolerance Approach
- Advanced Information Processing System (AIPS)
- Fault Tolerant Parallel Processor (FTPP)

BASELINE DESIGN AND VALIDATION PROCESS

- Baseline processor and network architectures for space station rely on a combination of computer self-tests and diagnostics and additional diagnostics by the crew to:

  - detect and isolate in-orbit failures of computers, networks and their interfaces to sensors and subsystems

  - reconfigure, repair and revalidate the system following each component failure

- Design and validation of distributed fault tolerant architectures is accomplished by extensive testing of ad hoc designs which have been put together using subjective criteria

NEED FOR VALIDATED FAULT TOLERANT ARCHITECTURES

- A validated fault tolerant computer system architecture is required that can autonomously:
  - monitor its own status
  - monitor the status of sensors and subsystems with which it interfaces
  - detect and isolate faults in computers, networks, sensors and subsystems
  - reconfigure the resources in realtime to continue to perform all the critical functions without manual (crew) intervention for reconfiguration and revalidation

- As Space Station evolves to support advanced missions, its computer system architecture must also extend to support:
  - mixed redundancy
  - function migration
  - reliable communication between physically distributed computers of different redundancy levels

DRAPER APPROACH TO FAULT TOLERANT COMPUTER DESIGN

- **Draper lab has designed fault tolerant computers which possess the following attributes:**

  - **validatable**
  - **Byzantine resilient**
  - **very low fault tolerance overheads**
  - **fault tolerance and redundancy management are transparent to user**

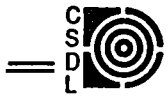
BYZANTINE FAULTS DEFINITION

- A Byzantine fault is an arbitrary behavior on part of a hardware component, a software module or a logical entity
- A particularly malicious manifestation of a byzantine fault is a "lying fault"

BYZANTINE RESILIENCE - MOTIVATION

Design Example:

- **Highly reliable digital control**
- **10e-9 per hour maximum system loss probability**
- **10e-4 per hour channel failure rate (typical)**



BYZANTINE RESILIENCE - MOTIVATION

Approach 1:

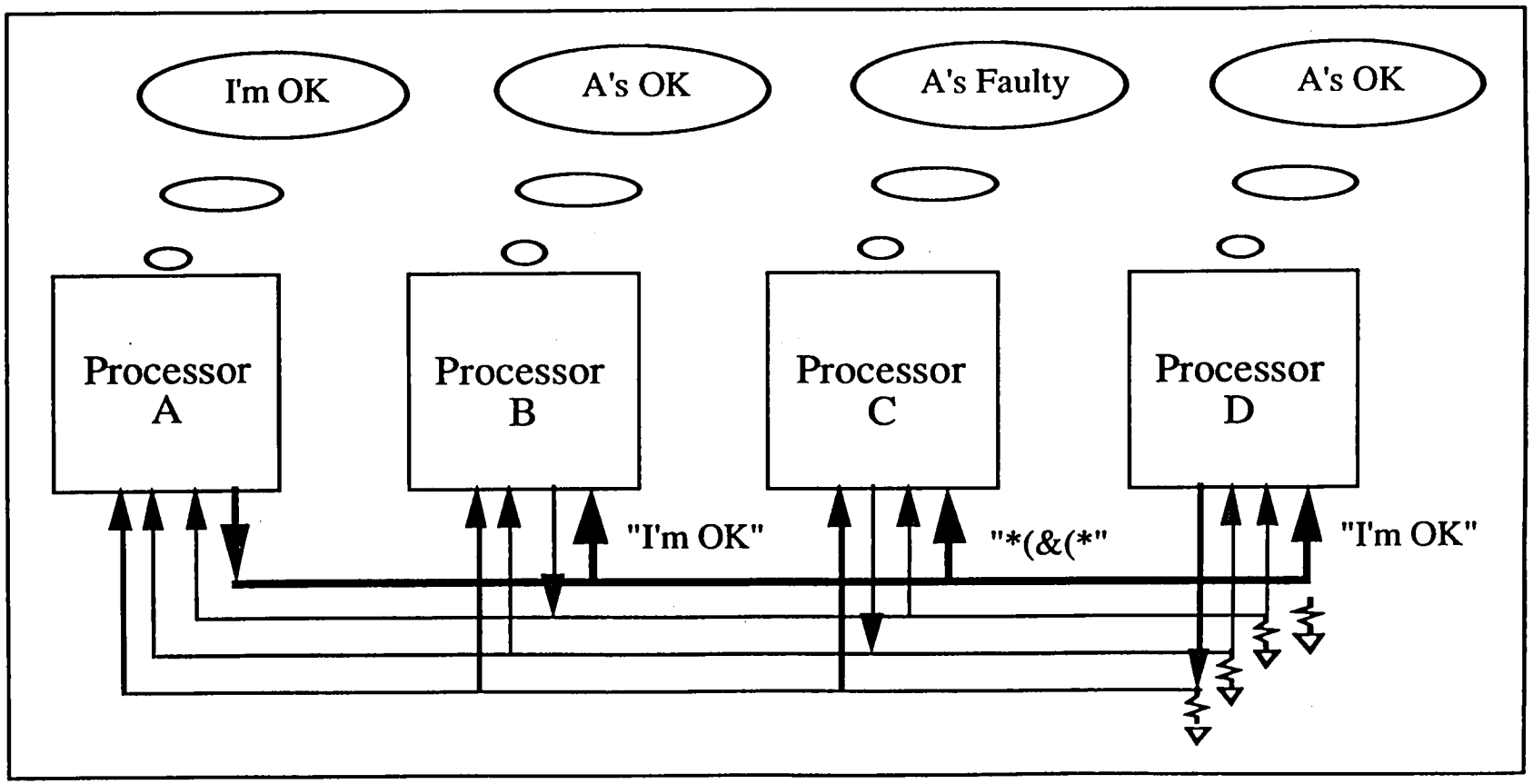
- Enumerate each possible failure mode and provide a fault tolerance technique for each
- Uncovered failure probability must be less than < 10e-5

Problems:

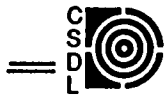
- Must demonstrate that fewer than 1 in 100,000 failures can cause loss of control
- Difficult validation problem
- Difficult reliability analysis problem

BYZANTINE RESILIENCE - MOTIVATION

Strange failure mode: Failure on A's transmit bus causes B, C, and D to diverge



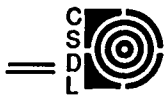
BYZANTINE RESILIENCE - MOTIVATION



Examples of strange failure modes:

- **In-flight failure of YC-14 flight control computer**
- **Promiscuous network node failure**
- **FTPP inconsistent vote errors**
- **FTP power supply-induced error**

BYZANTINE RESILIENCE



Approach 2:

- Make no guesses about likely failure modes
- Tolerate arbitrary failure behavior:

  "Byzantine Resilience"

Advantages:

- Theoretical requirements for Byzantine Resilience well-known and unambiguous
- Easier validation and analysis problem
- Higher reliability than other approaches

BYZANTINE RESILIENCE - REQUIREMENTS

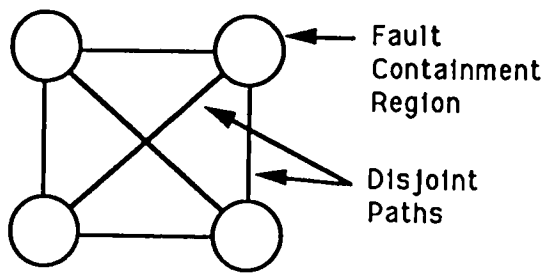
Theoretically correct implementation of Byzantine Resilience requires:

- **Bitwise comparison of results emanating from redundant sites of equivalent state complexity**
- **3f+1 fault containment regions (FCRs)**
- **2f+1 inter-FCR connectivity**
- **f+1 round inter-FCR protocol**
- **FCR synchrony**

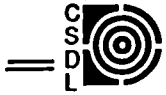
BYZANTINE RESILIENCE -REQUIREMENTS

1-Byzantine Resilient processing site:

- 4 identical FCRs
- fully connected
- 2-round interactive consistency protocol
- synchronous



ADVANCED INFORMATION PROCESSING SYSTEM



Objective

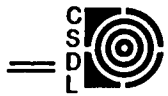
Develop an objective knowledgebase and demonstrate system building blocks which will allow achievement of validated fault tolerant distributed computer system architectures for a broad range of advanced aerospace applications.

Accomplishments

Demonstrated major attributes of the AIPS architecture using the engineering model:

- **Validatability**
- **Distributed computation**
- **Mixed redundancy**
- **Fault tolerance (processors, networks, interfaces)**
- **Damage tolerance**
- **Graceful degradation**
- **Expandability**
- **Transparency of fault tolerance to applications programmer**
- **Low fault tolerance overhead**

ADVANCED INFORMATION PROCESSING SYSTEM



Accomplishments

Produced an analytical and empirical knowledgebase for the validation of the AIPS architecture and building blocks:

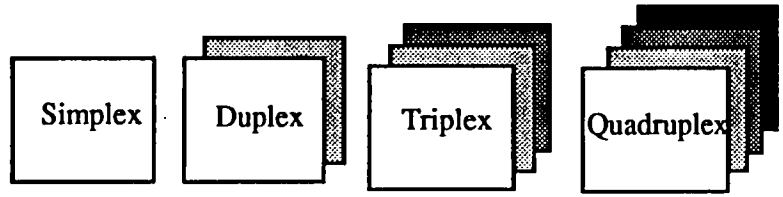
- **Architecture design rules and guidelines**
- **Analytical reliability and availability models**
- **Analytical performance models**
- **Empirical reliability data**
- **Empirical performance data**
- **Analytical cost models**

Completed the distributed engineering model for the demonstration of the AIPS attributes and for test & evaluation; Demonstrated the following buildingn blocks:

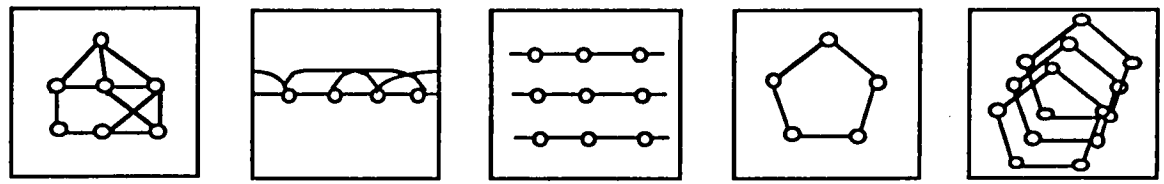
- **3 triplex FTPs**
- **1 simplex processor**
- **triplex intercomputer network**
- **mesh I/O network**
- **System Services software (85,000 lines of Ada Code)**

AIPS BUILDING BLOCKS: HARDWARE

FAULT TOLERANT PROCESSORS



INTERCONNECTION NETWORKS



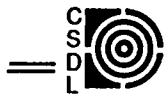
MESH | BRAIDED MESH | REDUNDANT BUSES | RING | REDUNDANT RINGS

INTERFACES

IOS Input/Output Interfaces

ICIS Inter-Computer Interface Sequencer

AIPS BUILDING BLOCKS: SOFTWARE



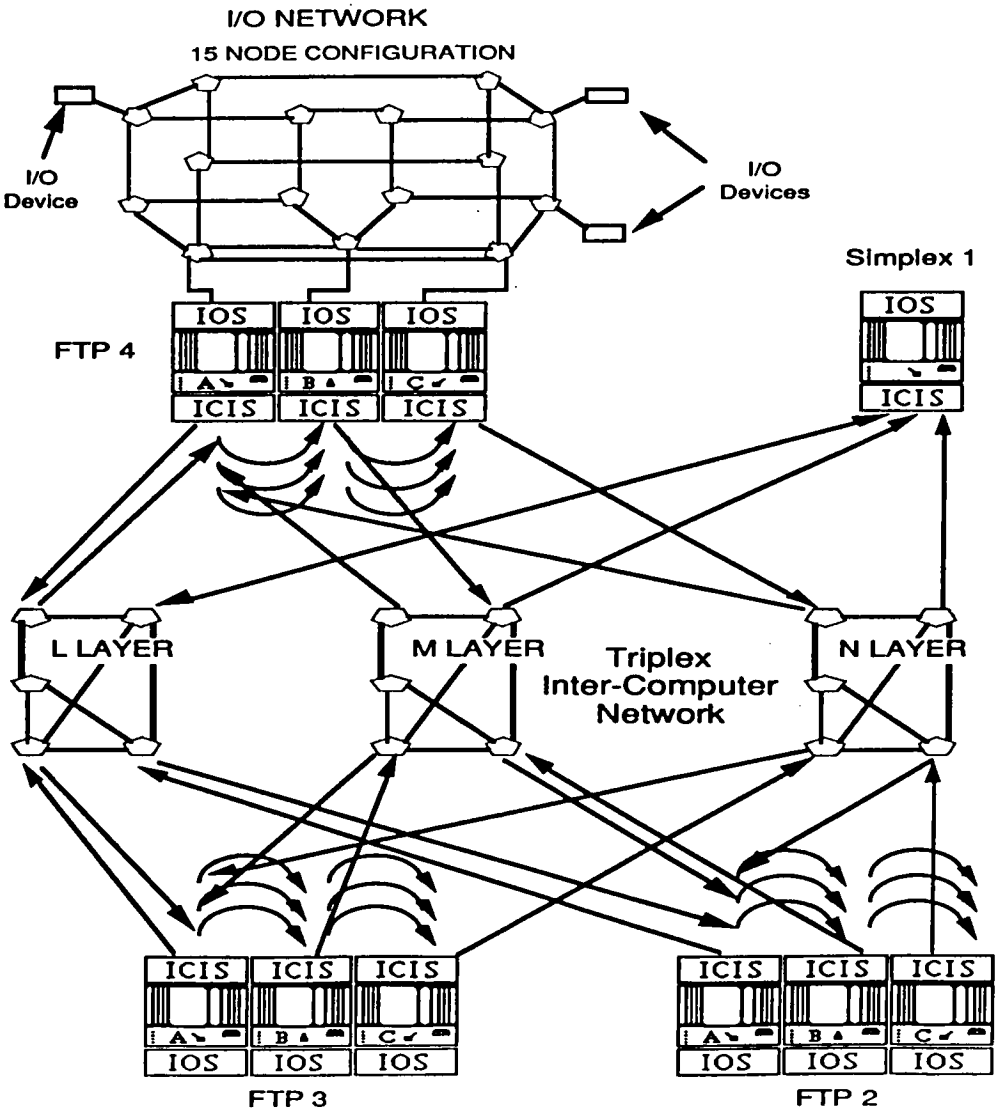
LOCAL SYSTEM SERVICES:
- Ada Real Time Operating System
- FTP Redundancy Management
- Local Time Management

INPUT/OUTPUT (I/O) SYSTEM SERVICES:
- I/O User Communications
- I/O Redundancy Management

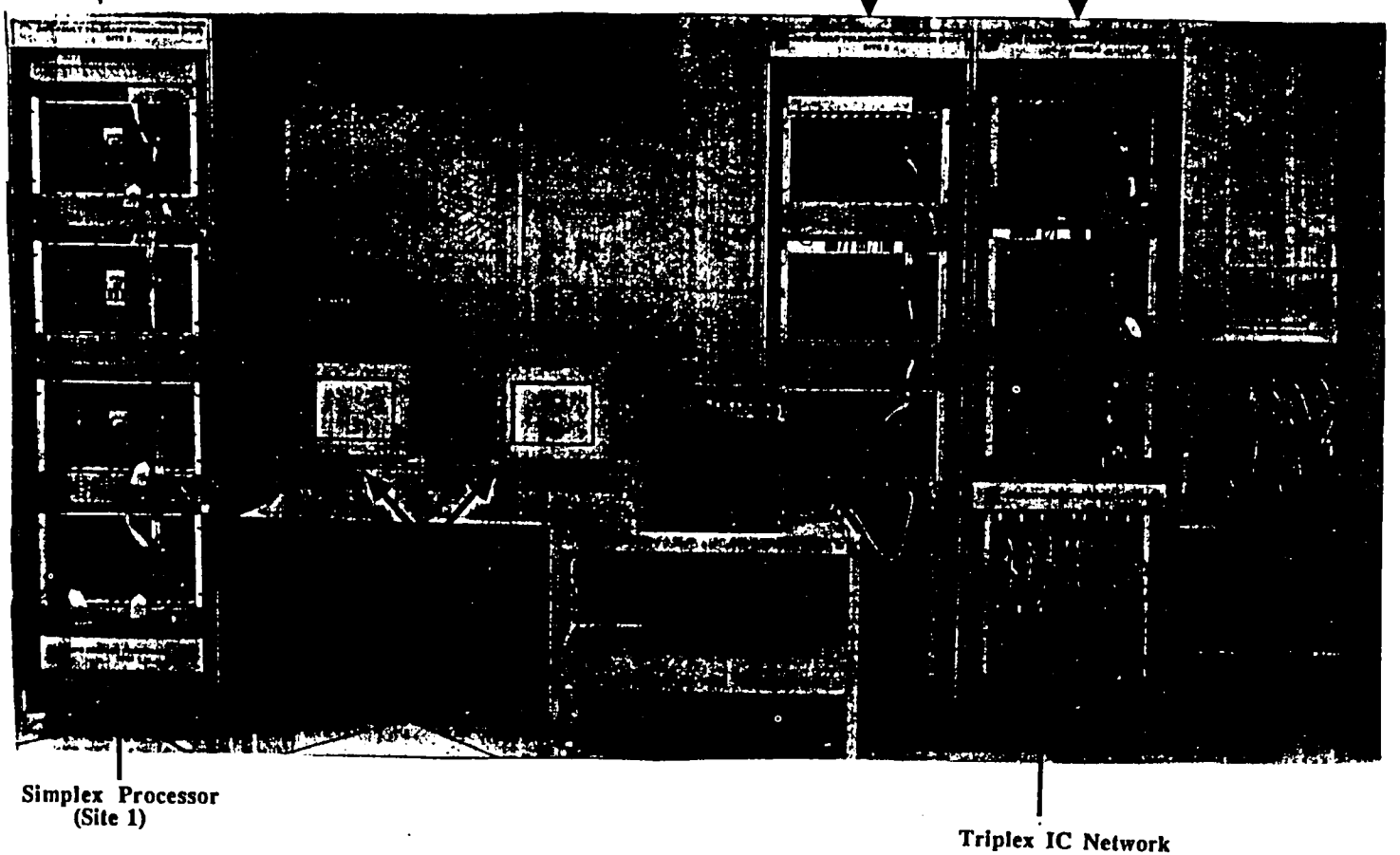
INTERCOMPUTER (IC) SYSTEM SERVICES:
- Ada Distributed Synchronous Communications
- IC User Communications
- IC Redundancy Management

SYSTEM MANAGER:
- Function Allocation & Migration
- System Redundancy Management
- Global Time Management

AIPS ENGINEERING MODEL CONFIGURATION



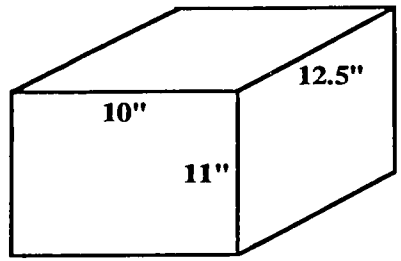
AIPS ENGINEERING MODEL



Q-2

TRIPLEX FTP CHARACTERISTICS

AIPS FAULT TOLERANT PROCESSOR



SIZE: 0.8 CUBIT FEET

WEIGHT: 50 lbs

POWER: 42 watts

THROUGHPUT: 20 MIPS

MEMORY: 4 MBYTES

RELIABILITY: 10^6 - 10^7 hrs MTBCF

PROB. OF FAILURE: 10^{-6} to 10^{-7} per hour

REDUNDANCY: Triplex (Expandable to quad)

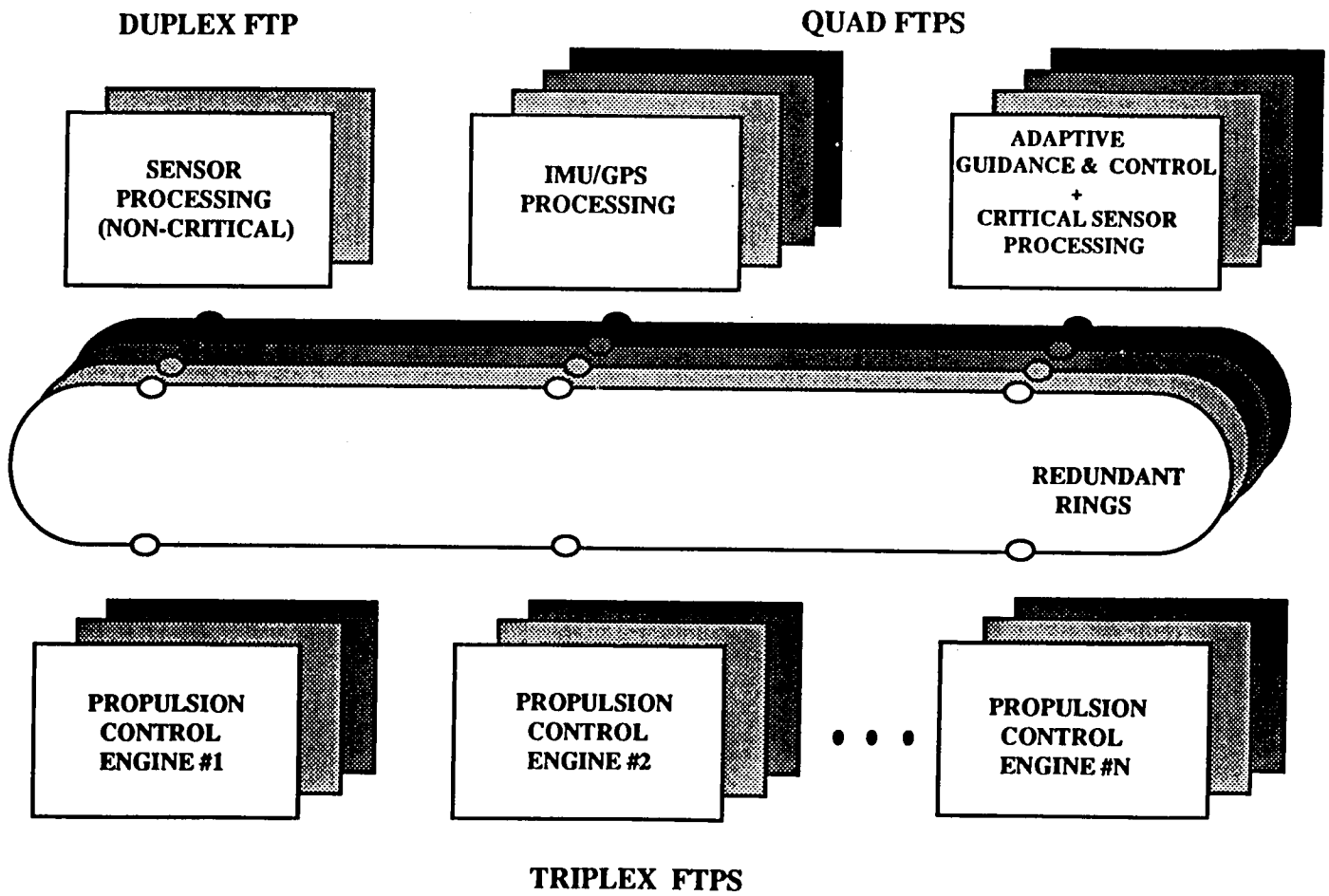
FAULT TOLERANCE: Fail-Operational, Fail-Safe

MODULARITY: Highly Modular

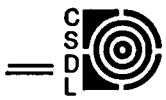
EXPANDABILITY: Network of FTPs

PROGRAMMING LANGUAGE: Ada

ADVANCED LAUNCH SYSTEM AVIONICS BLOCK DIAGRAM

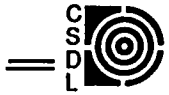


FAULT TOLERANT PARALLEL PROCESSOR OBJECTIVES



- Develop concepts applicable to high reliability parallel processing
- Demonstrate key concepts via proof of concept
- Make FTPP available to NASA/DoD community

ATTRIBUTES

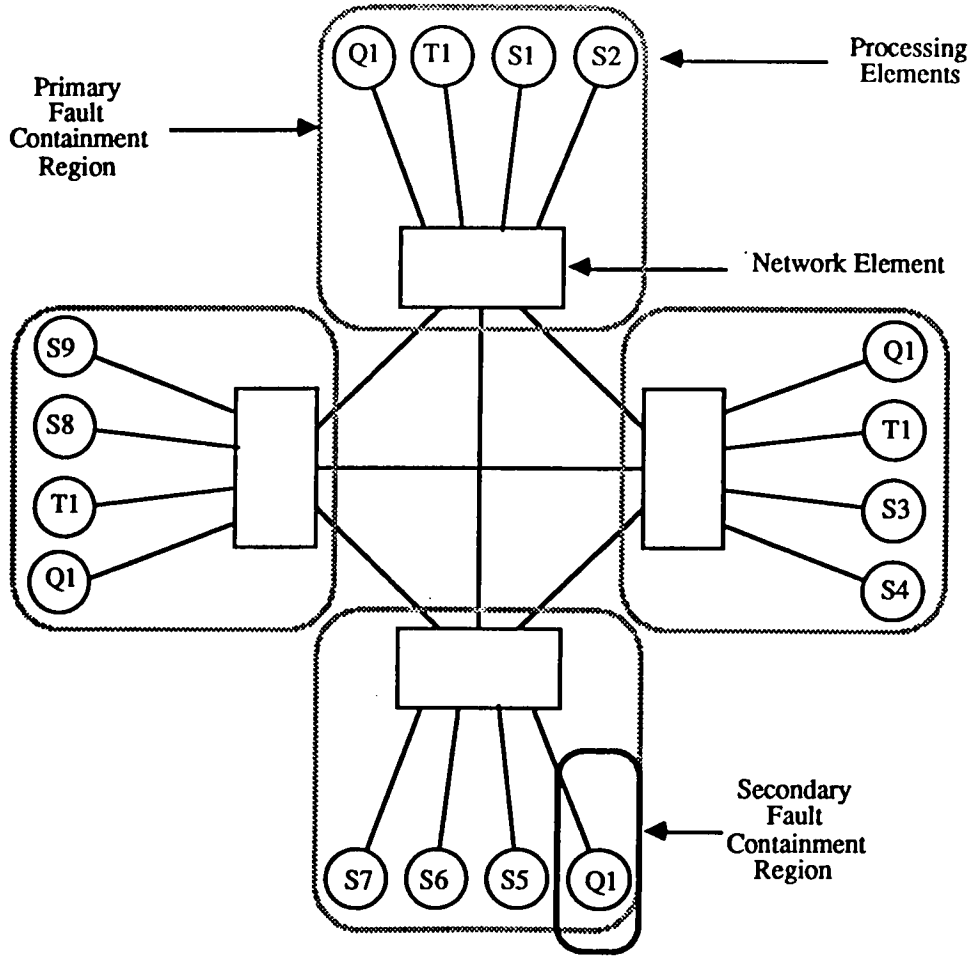


- **Message-passing parallel computer**
- **Messages serve both intercomputer communication and fault tolerant purposes**
- **Efficient hardware implementation of fault tolerance-specific functions**

ATTRIBUTES (CONT.)

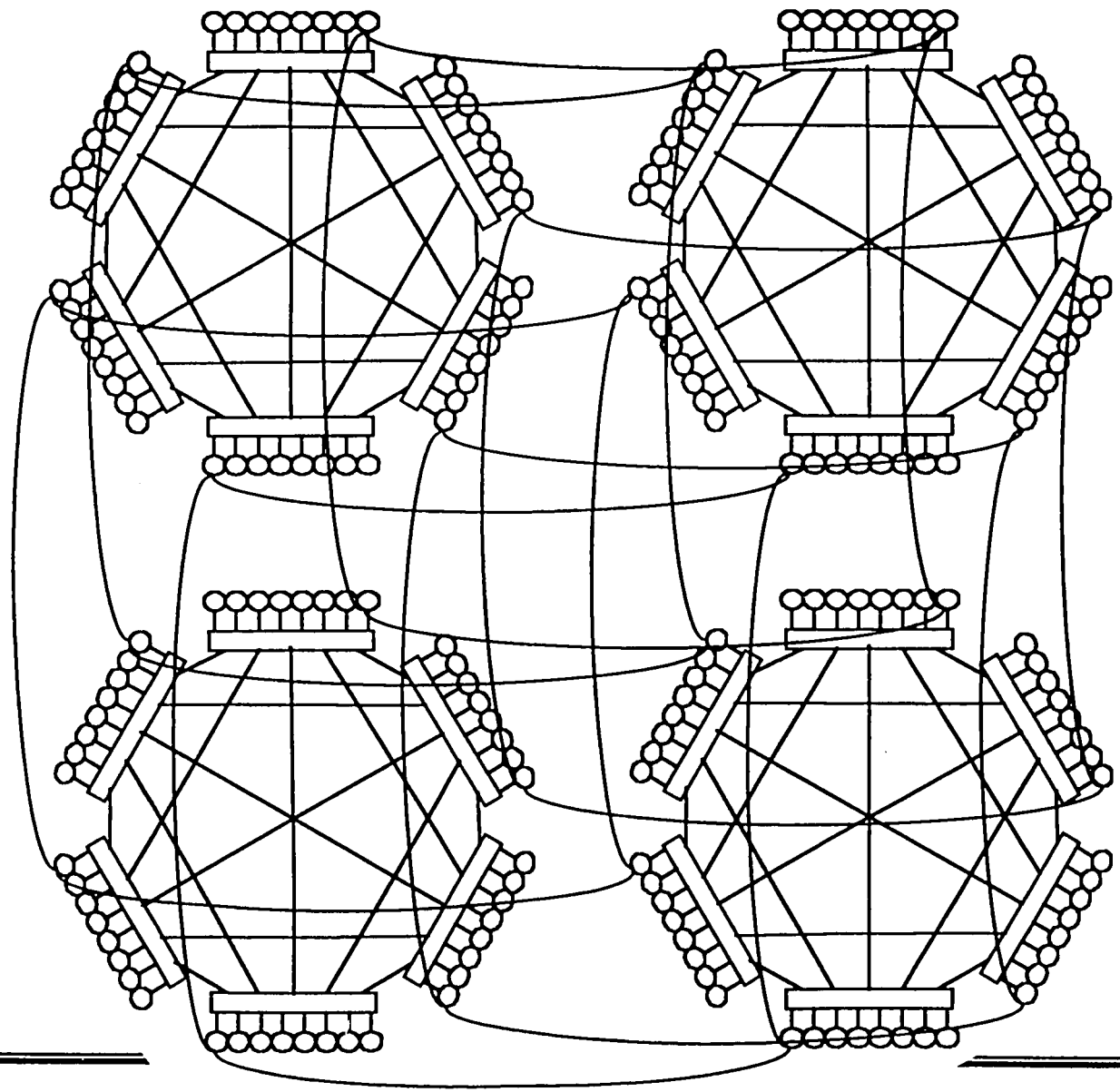
- Highly reconfigurable
  - Trades redundancy for throughput in real time
  - Allows variety of redundancy management modes
- Supports dissimilar processing sites
- Supports off-the-shelf components
- Programmable in C, ADA, or Assembler

FTPP POC CONFIGURATION



Q1 Quad 1
T1 Triad 1
S1-S9 Simplexes 1-9

192-PE FTPP



FTPP POC