

SSD-88-0044-E

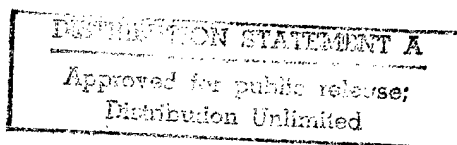
**STRATEGIC DEFENSE INITIATIVE (SDI)
SYSTEM ARCHITECTURE AND
KEY TRADEOFF STUDIES**

PHASE IIC

**CONGESTION CONTROL SUBSYSTEM
SOFTWARE ALGORITHM
APPENDIX E**

**22 MARCH 1988
CONTRACT NO. MDA903-85-C-0065**

19980513 043



NO QUALITY CONTROL

Satellite & Space Electronics Division
Rockwell International Corporation
2600 Westminster Boulevard
P.O. Box 3644
Seal Beach, California 90740-7644

PLEASE RETURN TO:
BMD TECHNICAL INFORMATION CENTER
BALLISTIC MISSILE DEFENSE ORGANIZATION
7100 DEFENSE PENTAGON
WASHINGTON D.C. 20301-7100

U2315

Accession Number: 2315

Publication Date: Mar 22, 1988

Title: Strategic Defense Initiative (SDI) System Architecture and Key Tradeoff Studies, Phase IIC: Congestion Control Subsystem Software Algorithm, Appendix E

Personal Author: Niznik, C.A.

Corporate Author Or Publisher: Rockwell International, Satellite & Space Division, Seal Beach, CA 90 Report Number: SSD-88-0044-E

Descriptors, Keywords: System Architecture Tradeoff Control Subsystem Software Algorithm Communication Network

Pages: 079

Cataloged Date: Dec 05, 1990

Contract Number: MDA903-85-C-0065

Document Type: HC

Number of Copies In Library: 000001

Record ID: 21250

SSD-88-0044-E

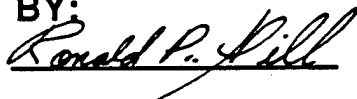
**STRATEGIC DEFENSE INITIATIVE (SDI)
SYSTEM ARCHITECTURE AND
KEY TRADEOFF STUDIES**

PHASE IIC

**CONGESTION CONTROL SUBSYSTEM
SOFTWARE ALGORITHM
APPENDIX E**

**22 MARCH 1988
CONTRACT NO. MDA903-85-C-0065**

APPROVED BY:



**Ronald P. Hill
Program Manager
SDI Architecture
Phase IIC Study**

Satellite & Space Electronics Division
Rockwell International Corporation
2600 Westminster Boulevard
P.O. Box 3644
Seal Beach, California 90740-7644

FOREWORD

The Satellite and Space Electronics Division (S&SED) of Rockwell International is pleased to submit this Final Report for Phase IIC of the Strategic Defense Initiative (SDI) System Architecture and Key Tradeoff Studies, in fulfillment of the requirements, as defined in the directed teaming arrangements, of contract MDA903-85-C-0065.

This report presents the results of the Rockwell team's efforts on this contract. These efforts were a continuation of the System Architecture and Key Trade-Off Studies initiated in December 1984, which Rockwell has been a part of since its inception. Rockwell's objectives in this study have been to refine analyses of previously developed architectures, and to develop and document new architectural variants. This effort resulted in the creation of technical data packages required to support the Defense Acquisition Board I decision and the development of the agreed upon baseline architecture documented in the Baseline Concept Description document.

This report is submitted in one volume with eight appendices.

CDRL A009	Final Report
Appendix A	Rockwell SDI Architecture Study Document Index
Appendix B	Operations Concept (CDRL B003)
Appendix C	Architecture Evaluation (CDRL B006)
Appendix D	IDEF ₀ Function Decomposition Flows
Appendix E	Congestion Control Subsystem Software Algorithm
Appendix F	Automated Systems Engineering (EPOS)
Appendix G	AERIE System Concept
Appendix H	Manufacturing Assessment

The cost estimates set forth in this document are submitted for budgetary and planning purposes only and do not constitute a firm commitment on the part of Rockwell International Corporation.

PREFACE

This appendix contains a congestion control subsystem software model and its supporting theories. The model presents data and algorithms for avoiding deadlock, livelock, and other problems associated with communications between space-based nodes through a hostile environment. This document was generated by Dr. Carol A. Niznik of NW SYSTEMS. The analysis was performed under a subcontract from Rockwell International for Phase IIc of the SDI Architecture and Key Trade Offs Study.

TABLE OF CONTENTS

<u>Section</u>	<u>Title</u>	<u>Page</u>
	Foreword	i
	Preface	ii
	Table of Contents	iii
	Illustrations	iv
1.0	Introduction	1
2.0	Overview of CCSCR Theoretical Model Concept Problem	3
3.0	Contention Resolution Sequencing Algorithm	4
4.0	Message Path Load and Propagation Delay Monitoring Impacts on Sequencing Subalgorithm Concept	42
5.0	Update Information Interpreter Subalgorithm Development	54
6.0	References	60
	Acronyms and Abbreviations	65

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Page</u>
1 Overview of Sequential Contention Resolution Algorithm Theoretical States and Their Interaction	7
2 Overall Flowchart of Contention Resolution	8
3A I/O for Computer Network Nodes	9
3B CCSCR Algorithm Software Imbedding	9
4 Deadlock and Livelock Prevention Subalgorithm	16
5 Satellite Model for Deadlock-Livelock	18
6 Resident Time Subalgorithm for m Node Network Fragments	20
7 Global Residence Time in a Star Network Subalgorithm	21
8 Calculation of Residence Time in a Ring Topology	23
9 Real-Time Parallel and Sequential Input Buffer Arrangement Subalgorithm	28
10 Number of Requests in a Ring Network Subalgorithm	29
11 Load Calculation Subalgorithm at Each Node in the Network	31
12 Load Calc Subalgorithm	32
13 Matrix A	33
14 Number of Requests at All Nodes in a Star Topology	35
15 Buffer Resolution Subalgorithm	36
16 Race Condition Recovery Algorithm	40
17 Update Information Interpreter Subalgorithm	57

1.0 INTRODUCTION

The fundamental requirement that must be satisfied by the SDI BM/C² Communication architecture is to provide a transport interface to each of the three principal SDI functions: Surveillance (sensor subsystems), Battle Management (Sensor, Resource and Weapons Battle Management subsystems), and Weapons (launch and interceptor subsystems). This requirement for transparency can be realized through the integration ("tight" coupling) of two subarchitectures that comprise the overall SDI BM/C² Communication architecture. Within each node (satellite), a data flow control architecture (DFCA) can be developed from the node's sensor subsystem to its end-users (SBMs) to provide the capability to transmit a very large volume of track file data (these track files do not consist of state vectors) at high throughput rates, in a congestion-free (prevention of deadlock, livelock, and thrashing conditions) manner. The node's DFCA can then be coupled to the inter-node data link network architecture (IDLNA) to form an integrated, disciplined and synchronized communication system with capability of providing a continuous (uninterrupted flow of track file data, at very high speeds, from any sensor subsystem to its corresponding SBMs (SBMs based at the host and stereo-viewing nodes). The DFCA addressed in this report is based upon satisfying the aforementioned requirements and the following mission-performance related conditions:

- 1) The availability in Real-Time to all end-users (SBMs and RBMs) of target/object track file data;
- 2) Uninterrupted flow of data from the sensor focal plane to its intended end-users (SBMs and RBMs);
- 3) A high reliability of communicating consecutive frame cycle track file data to the appropriate SBMs for stereo-processing;
- 4) Synchronous and asynchronous parallel, simultaneous, transmission of target/object track file data from each sensor's subsystem to their corresponding end-users (SBMs).

The real-time condition is based on satisfying the critical timeline requirement which dictates that the total time allocated between target/object detection and weapon activation (launch of SBI) must be ≤ 1 percent of the sensor frame cycle time. Inherent in this timeline constraint is the need to satisfy target/object track prediction accuracies. Conditions 2, 3, and 4, above, are necessary to ensure that the critical node outages are prevented from occurring and essential data is available, in a timely manner, to the SBM's tracking filters for accurate track trajectory predictions.

The synchronous Congestion Controlled Sequential Contention Resolution (CCSCR) algorithm analyzed in this report addresses all of the principal concerns mentioned above. The CCSCR algorithm incorporated in the interface subsystems within each SDI satellite can provide the mechanism for realizing the requirement of congestion-free communications throughout the SDI BM/C² Communication System.

2.0 OVERVIEW OF CCSCR THEORETICAL MODEL CONCEPT PROBLEM

The specific problem that the CCSCR (Congestion Controlled Sequential Contention Resolution) Software Algorithm, derived and developed in this contract, solved was the efficient, deadlock and livelock free processing of large numbers of simultaneous parallel input track files in a manner enabling their time planned simultaneous parallel output. The transparent, sequential, serial and combination serial, and parallel race condition free solution offered by the CCSCR Algorithm can be employed in ground- and space-based communication network gateways and computer node interface subsystems within each of the three Strategic Defense System satellites: BSTS, SSTS, and SBI. This solution can be employed in the Phase IIC SDI Communication Architecture Systems. This report describes the linking software enabling the battle manager and the network manager to communicate through an interface in a congestion effective manner. In the case of a fast initial boost, the software theory will have to be flexible enough to work within a shorter time cycle incorporating a reduced number of phases.

3.0 PROBLEM STATEMENT: CONTENTION RESOLUTION SEQUENCING ALGORITHM

Contention resolution will be addressed in a parallel input and output network by the treatment of the inputs as asynchronous, separately buffered inputs that enter and leave the network in a sequenced manner.

Problem Solution

Abstract

The synchronous Congestion Controlled Sequential Contention Resolution (CCSCR) Algorithm developed here for parallel simultaneous, or asynchronous input and output networks, was motivated by the development of a network operating point, where no livelock or deadlock will occur because of the inclusion of the preventive protocol techniques of the varying sequencing of message service, buffer storage arrangement, and rearrangement. The following three objectives were applied to the development of all possible features of this CCSCR Algorithm: (1) Identify network segments where fragmentation, heavy load, and topological variations and combinations occur under real time conditions. (2) Develop the software theory for linking these algorithm sections together and develop the main contention resolution sequencing algorithm. (3) The two additional artificial intelligence theoretical characteristics of self organization and optimal stopping are included in the sequencing algorithm to enable the determination of the length of time for a routing sequence in the network, and a theoretical development of the time when a message transit stops prior to its normal stopping time. This optimal stopping time determination will occur due to external networking conditions requiring the temporary storage of the present message transit parameters and simulated performance.

This CCSCR Software invention provides the mathematical circuit control concepts required at three computer nodes or gateway computer nodes hardware levels; i.e., (1) front end processor level, (2) operating system programming level and (3) operating system arithmetic level, to enable the congestion controlled optimal transmission operation of circuitry at these three

computer nodes or gateway (bridge) computer nodes by working in a series of sequentially software motivated hardware operations that for all functional purposes appear to be handling their input link data in a simultaneous parallel manner. Therefore, in the CCSCR Algorithm, the sequential circuit operation of the three levels which allows simultaneous network input acceptance is transparent to the other sections of hardware and inputs to the network nodes.

The areas of deadlock and livelock, i.e. various forms of saturation of the computer network which enable a congested condition to occur, are prevented in contention resolution algorithms by appending a monitor of the network load for a real-time environment at the times indicated by Self Organization Theory. Self Organization Theory predicts network message phase cycle length and the optimal determination of network function stopping time due to external network conditions. The computation of these functional network cycle lengths and function stopping times by self organization theory and optimal stopping theory are enhanced by message residence time computations from monitored nodal and gateway service rates, interarrival rates and buffer storage allocation. The topological considerations of rings, stars and their combinations, are dealt with in conjunction with the network overall and individual nodal and gateway real time loads.

CONTENTION RESOLUTION SEQUENCING ALGORITHM

I. INTRODUCTION AND OVERVIEW

The areas of deadlock and livelock, i.e. various forms of saturation of the computer network which enable a congested condition to occur, are prevented by the use of contention resolution algorithms in intra-node communication network (ICN) interface subsystems that control the flow data between computers. The CCSCR Algorithm employs a monitor that determines the load at various nodes within the network in real time. This information is used by each of the processors. The contention resolution algorithm is programmed into these processors as well as into the operating system as illustrated in Figure 3B. The self organizing theory predicts network message phase cycle length and the optimal determination of network function stopping

time, due to external network conditions. The computation of these functional network cycle lengths and function stopping times based on self organization and optimal stopping theories are enhanced by message residence time computations from monitored nodal and gateway service rates, interarrival rates and buffer storage allocation. The topological considerations of rings, stars and their combinations are dealt with in conjunction with the network overall and individual nodal and gateway real time loads. Refer to Figure 1 for a generalized overview of the linking order of these theoretical network state principles within a contention resolution sequencing algorithm. Figure 2 illustrates the more detailed approach to the description of the network states. The algorithm for the linking of the contention resolution software to these other features is detailed in Figure 2.

The purpose of this overview is to treat the contention resolution problem in the SDI/Architecture Satellite networks where there could be a maximum of M inputs simultaneously to their respective buffers. Since the incoming message rate can be in the gigahertz range, and there may be the possibility of fewer than M transfer processors available between the input and output buffers, data from the input buffer will compete for the processors. Refer to figure 3A for a diagram of the layout of the SDI/Architecture problem.

Refer to Figure 1 for a generalized overview of the linking order of the theoretical network state principles within this contention resolution sequencing algorithm. Figure 2 illustrates the more detailed approach to the description of the network states within a computer network test bed algorithm. The algorithm for the linking of the contention resolution software to these other features is detailed in Figure 2. Referring to Figure 3A, when there are a maximum number of M parallel (simultaneous) inputs to a computer node front end processor, and this node has M parallel input lines linked to computer node buffers, there could be less than M processors or M available output links resulting in a conflict or contention problem. The messages can enter the computer node/s buffers incorporating two different methods of control for contention resolution of simultaneous synchronous or asynchronous message inputs to occur. Refer to Figure 3B for a diagram of the

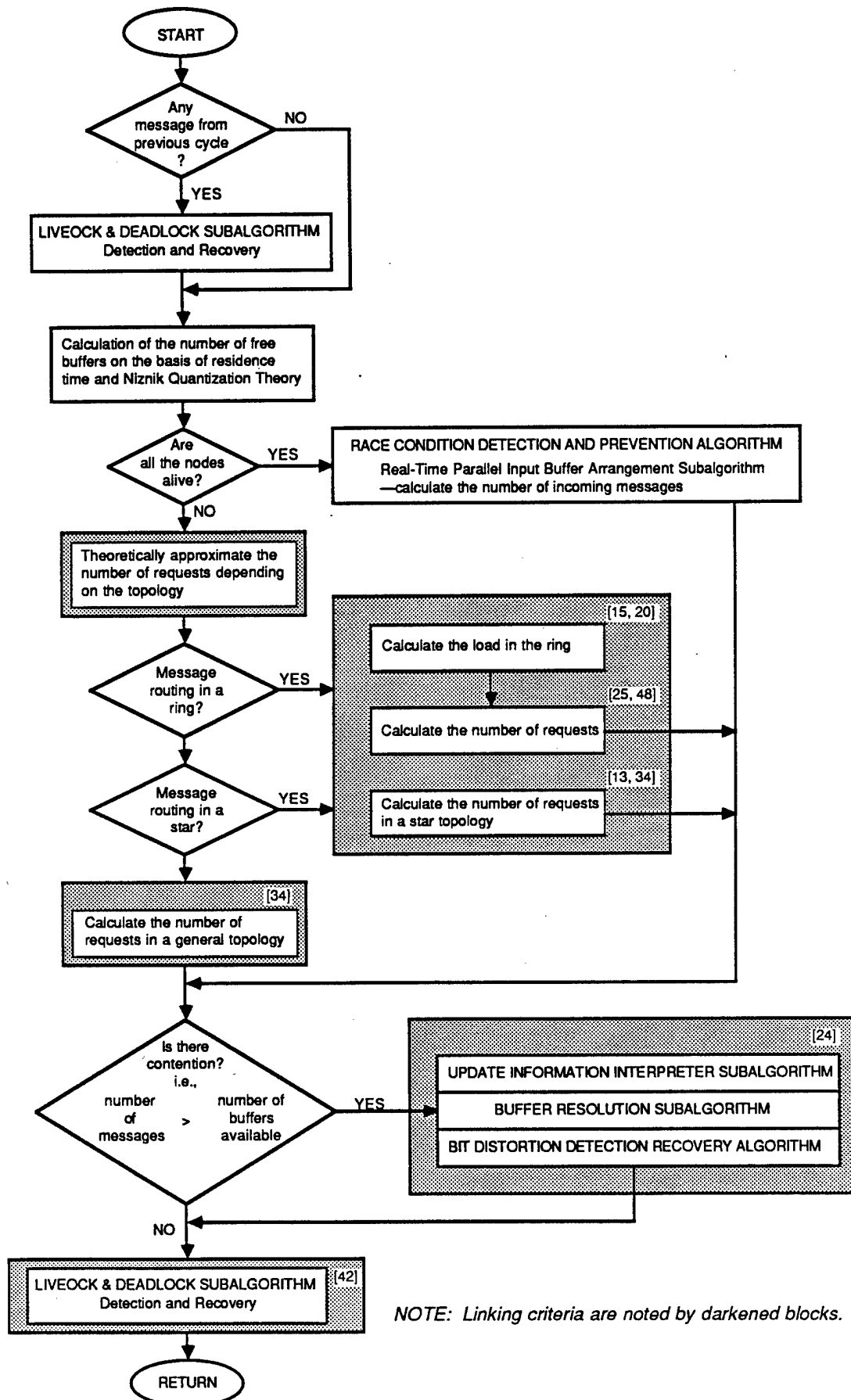


Figure 2. Overall Flowchart of Contention Resolution

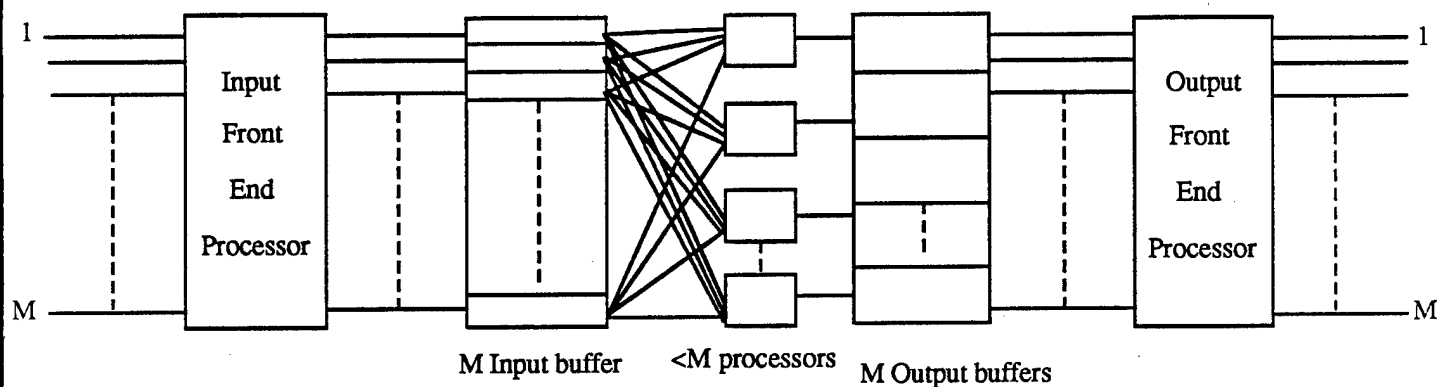


Figure 3A. I/O for Computer Network Nodes

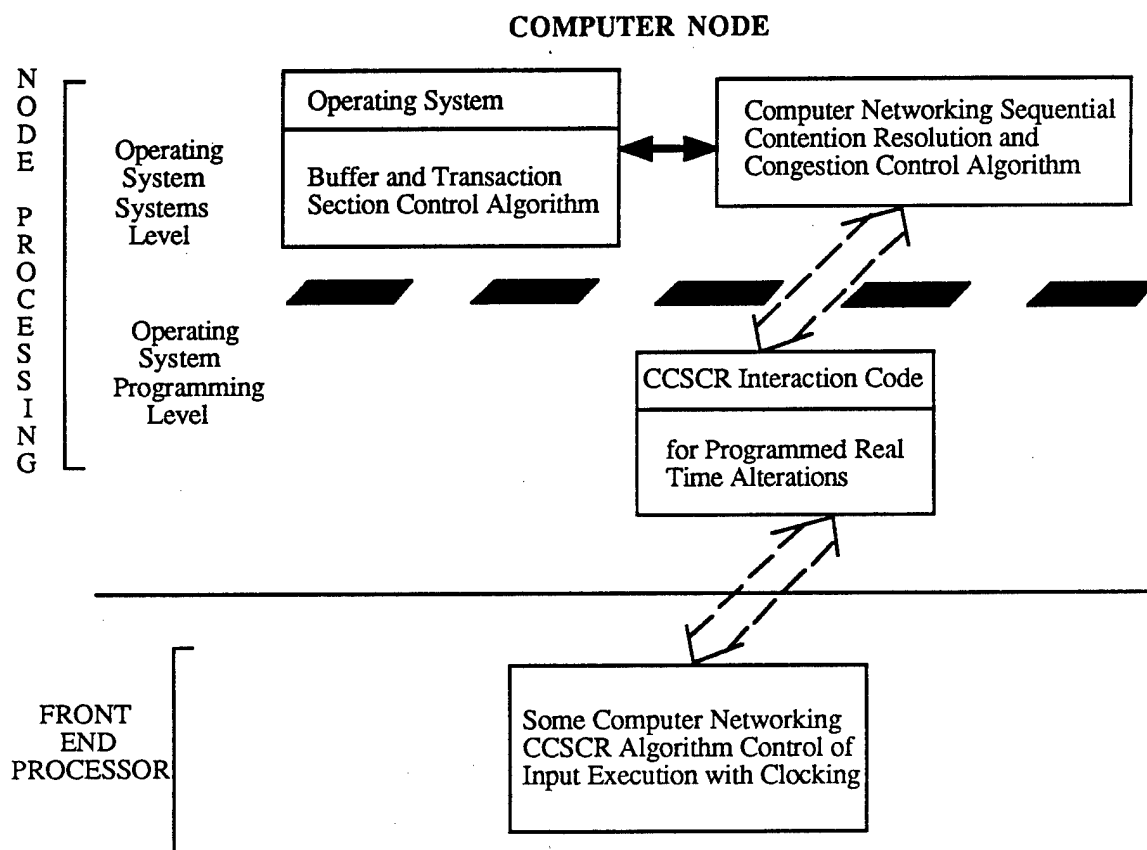


Figure 3B. CCSCR Algorithm Software Imbedding

operating system layout of the solution of this problem with the CSSCR Algorithm.

II. GENERAL SEQUENTIAL CONTENTION RESOLUTION PROBLEM STATEMENT FOR SDI ARCHITECTURE

Referring to Figure 3A, when there are a maximum number of M inputs to a satellite node, and this node has M inputs lines linked to computer node buffers, there could be less than M processors or M available output links thereby resulting in a conflict or contention problem. The messages can enter the satellite computer node/s buffers as simultaneous synchronous or asynchronous inputs. This conflict and contention for resource problem can readily result in a condition of saturation at both the computer and the interface buffers. Saturation produces a state of deadlock, which generally becomes irreversible. A communication outage occurs for the data link containing these computers/buffers. The surveillance and weapon satellites (BSTS and SSTS) are the nodes that are vulnerable to this problem.

III. CONTENTION RESOLUTION SEQUENCING ALGORITHM SOLUTION FOR SDI ARCHITECTURE

A. Buffer Input Arrangement and Departure Process Solution

Simultaneous parallel inputs, whether they are synchronous or asynchronous should be dealt with by their placement in the computer node buffers, in a sequenced [5,7], grouped, permuted, or singular manner. This concept enables the computer node to process them in a manner which appears conceptually simultaneous (parallel) in real time, but is factually sequential, because of the order in which these messages or their sections are processed.

1. Automaton Formalism Solution To Parallel Input Contention

The automata that realize parallel congestion control can exist in more than one networking processing state at a time. In addition, the CCSCR Algorithm synchronizes the input and output buffers. The innovative concept here is that less than M transfer processors can be used to perform the status and race condition assessment function of a network that contains M parallel inputs and outputs. Each processor serves to analyze the outputs from all buffers simultaneously in order to determine if a deadlock, livelock, or race condition is developing. Each processor forms an M parallel to serial conversion at its front end. Then, its CPU compares each message sequence with respect to a criteria that can be maintained if a race condition occurs. When the synchronous input automaton is used to implement this sequential control algorithm, a minimum length nonorthogonal* code [7] for simultaneous parallel states and an orthogonal** [7] code for non simultaneous parallel states is represented in the automaton model developed by Cheremisinova [7]. For the asynchronous input automaton model, the solution to the race conditions (between memory elements), minimum length nonorthogonal codes for parallel states and orthogonal codes for nonparallel states, must be used in the automaton formalism. Refer to section E for the further race condition definitions and problems solutions.

(a) Antiracing state assignment for asynchronous nonsimultaneous inputs.

In this case, a method for creating an antiracing state assignment based on direct network state transition is employed. The assignment variable creates a two-block partition (S_1, S_j) of states where the transition is from the set of states in S_1 to the set of states in S_j . During this assignment, the racing between memory elements can be eliminated based on the input state and a few conditions.

(b) Antiracing state assignment for asynchronous simultaneous inputs.

At each epoch in time, a parallel (simultaneous input) automaton is simultaneously in all the parallel states represented by the input to the network or the computer node or gateway. A transition is defined from a set of parallel states. The transition involves the change in value of different or the same variables. If the changes are for different variables then these state transitions can occur in the parallel simultaneous input mode. However, if the changes are for the same variable, then there is racing for the set of nonparallel states. To eliminate this race, the input state and a few constraints have to be considered to eliminate the race conditions. (Refer to the Race Condition Detection and Prevention Algorithm, Figure 16, and Section G). Therefore, the state assignment matrix for the automaton must be determined, such that it satisfies the condition for orthogonality for nonparallel, i.e. nonsimultaneous states and the condition of nonorthogonality for parallel, i.e. simultaneous network states.

* Nonorthogonal code for simultaneous input states - the rows of the resultant assignment matrix has no opposite values in columns corresponding to parallel states. A common implicant, i.e. ternary vector, implicates these rows.

** Orthogonal code - the rows of the resultant assignment matrix have opposite values in columns corresponding to nonparallel states.

2. Critical Section And Parallel Memory Update Problem In Computer Network Distributed Databases

In distributed databases where the same database resides in multiple nodes of a network, the problem of parallel memory update is critical to the performance of the network. Transferring to Figure 3B, if a particular transaction updates the database in one node, then this effect should be reflected in all the nodes where a copy of the database resides. This is the problem of synchronization and some methods for solving it are (a) naming and (b) locking [33]. Another problem is that of the critical section at the simultaneous, synchronous input node or network processes. This problem is represented by two or more processes attempting to update the value of the same variable in the distributed database simultaneously. The result would then be an irrelevant value for the variable. Several solutions to this problem use the serialization of the processing of data, assuming that only one of the processes will obtain access to the critical process section, where the messages are processed in series. It is practical for several processes to obtain access to the critical process section.

(a) The Problem of Concurrency in Centralized and Distributed Databases.

If many processes access the same centralized database [32] then we have the updating and concurrency problem. If these processes run in parallel and have access to the same database, and attempt to update the same variable in the database, then there is a concurrency problem. To solve this problem, a scheduler is used. This scheduler shuffles the job in such a manner that the correctness of the variable is maintained. A digraph of the processes is developed which depends on their interrelationship to each other. In order to be conflict serializable, i.e. the conflicting parallel processes should be serialized in the order of arrival, the digraph should be acyclic. The problem of concurrency is solved by a variety of methods. The usual way to solve this problem is by using semaphors, which is a software solution, or by test and set which is a hardware solution. There is a later reference to multiphors in Section V.

The problem discussed above was for centralized databases. The problem of concurrency becomes more serious in the case of distributed databases [36]. In this case the same database is distributed over different geographical locations. Strict concurrency control has to be exercised in this case. All processes should use the current database. To ensure this, we should linearize the conflicting processes and if a process is aborted or terminated we should be able to roll back, i.e. restore the database to its previous state. We can also have deadlocks in this case. The algorithm for concurrency control for distributed databases as proposed in [36] takes care of the deadlock and the concurrency problem. The algorithms are wound-wait and wait-die. In both of these algorithms, in order to have concurrency control, we abort and restart the processes depending on specific conditions.

B. Front End Processor Additional Capability of Sequential Entry of Messages To The Buffer And Departure Of Messages From The Buffer Solution

This is a form of buffer message arrival rate and service rate ordering or enabling [2,24,43]. This front end processing sequential run through, takes into account the previous history, i.e. if the station was given access in a previous "selection" then it would be given a chance to transmit in the present "selection". This algorithm ensures limited access delay. The selection process described is a form of front end processing for the central node where decisions are made. The front end processor has this serial capability as well as the parallel, simultaneous processing capability.

C. Deadlock And Livelock Control Subalgorithms

Deadlock and livelock are network states which occur as a result of contention or saturation. The basic contention resolution sequencing algorithm created is overlayed in a hierarchical manner [15] on a subalgorithm which will prevent network node congestion from occurring that can result in deadlock and livelock. Refer to Figures 1 and 2 for the location of the deadlock and livelock software controllers in the CCSCR Algorithm. (Refer to Figure 3B for the software and hardware gateway computer node and computer node location of the CCSCR Algorithm.) Computer network deadlock [42] is defined in a

situation in which scheduling of packets dynamically prevents one or more packets from reaching their destination. Computer network livelock is the condition where the messages still maintain a cyclic motion within the network, however the messages never reach their destination. For nodes with the highest degree of connectivity, a livelock-deadlock controller, i.e. congestion control will insure that a datablock reaches its destination after a finite amount of time, i.e. that the responded time is controlled. A FSET (Forward-State-Elapsed-Time) serviced-instant generated deadlock and livelock controller was used in reference [42] to permit or prevent various message transits in the network. Local nodal controllers, where each processor decides on the legality of accepting a packet from a local state of a node v are the following parameters:

$$J = (j_0, j_1, \dots, j_k)$$

where j_i , ($0 \leq i \leq k$) is the number of packets in V , whose destination distance from v is i and

$$T = (t_0, t_1, \dots, t_k)$$

where t_i ($0 \leq i \leq k$) is the earliest, i.e. the smallest, creation time among the packets waiting to be accepted by v , whose distance from v to their destination is less then or equal to i . If no packets of this type occur set $t_i = \infty$.

Any packet P requesting access to a node v has a packet state relative to v characterized by the following parameters:

j = the distance from the node v to the destination of P and

t = the creation time of the packet P .

A conceptual flowchart of the FSET is described in Figure 4. The FSET is defined for $b > k$ when (b, k) is the following set:

$$(b, k) \in ([J, T], (j, t)] \mid 0 \leq j, d^* = d(J) \text{ and } 0 \leq t \leq (t_{a^*-1})$$

where b is the number of buffers in each node and k = length of the longest path in the network. The distance threshold in Figure 4 is derived for state $j = (j_0, \dots, j_k)$.

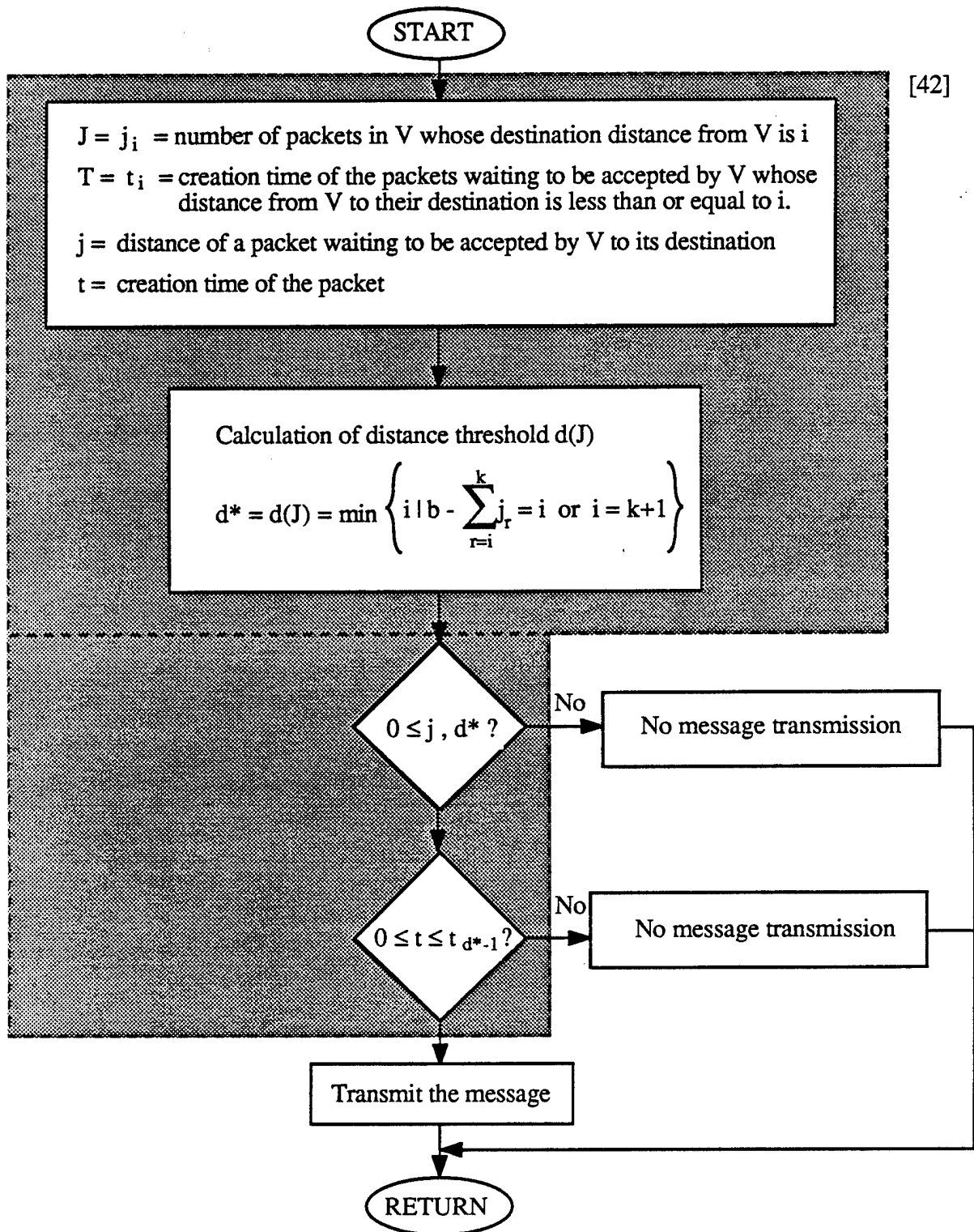


Figure 4. Deadlock and Livelock Prevention Subalgorithm

1. Livelock/Deadlock Example In A Satellite Network

Referring to Figure 5, node N_2 is the satellite node and all other nodes are ground nodes. If each node including the satellite node has 10 buffers, then the messages are routed from N_1 to N_{10} and W_1 to W_3 through N_2 . To prevent deadlock during transmission between N_1 and N_{10} , there must remain 9 free buffers at N_2 , 8 free buffers at N_3 , etc. To prevent deadlock for transmission between W_1 and W_3 there must be 1 free buffer at N_2 .

Therefore, under a heavy load at W_1 , there will be an imbalance soon. Node N_2 is more likely to have 1 free buffer instead of 9 free buffers. In this case of a livelock example, messages from W_1 will always be transmitted and messages from N_1 will keep waiting for the infrequent opportunity of 9 free buffers occurring at node N_2 . In this case messages from N_1 will never reach their destination. This livelock condition can be solved by simulating the FSET protocol at N_2 . Since this is a livelock condition, messages continue to flow however, they never reach their destination.

D. CCSCR Networking Parameters

1. Residence Time.

In order to obtain the number of free buffers at any point in time in the network, the residence time, (the time a message spends in network fragment, i.e. subnetwork section), must be determined. The topology of the network can be considered to be a ring, star or the nodes in a subnetwork or nodes in a network path. The nodes in the fragment can have either an interactive or passive state. The residence time in a fragment increases as the number of interactive nodes in the fragment increases. The nodal or gateway interarrival and service rate determines the time a message exists in a fragment. A network fragment [37] consists of k nodes of a network, I , comprised of N nodes, where $N > k$, $I = (1, 2, \dots, N)$ and $k \subset I$. The residence time of the process in node k over the interval $[0, t]$ for $T_{i,k}$, where i is the initial node. The Accumulative Residence Time in a subset of Nodes $k \subset I$ (k is a subset of the fragment) is,

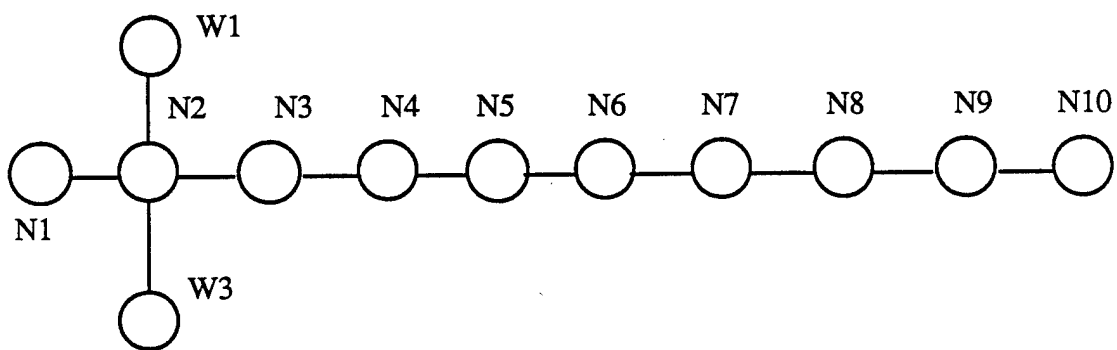


Figure 5. Satellite Model for Deadlock-Livelock

$$\pi_{i,k}(t) = \sum_{k \in I} \tau_{ik}(t) \quad (1.1)$$

Several systems of linear differential equations (SLDE) will have to be solved to form the solution equation, i.e. the Kolmogorov SLDE. Refer to Figure 6 for the flowchart solution to the residence time for n computers which can be either actively or passively served. Here, a terminal goes into an interactive state with service rate intensity μ and remains in this state for a time which is exponentially distributed. The conceptual formalism solution in the Figure 6 flowchart indicates the solution to the residence time when there is more than one terminal in an interactive state. The Kolmogorov SLDE is described in the flowchart of Figure 6. Refer to Figure 7 for the flowchart of the residence time computations for the star network topology [13,15,22,27].

The star network topology was formally defined in terms of the parameters described in Figure 7. The specific star network model developed for residence time computation requires that the requests can be served by an SUCS* Computer. These computers have the same intensity μ and there are g computers, one with traffic intensity η . The transition probabilities Q_{ij} are given by the following equations for $g=3$,

$$Q_{12}(t) = 1 - e^{-\lambda t}$$

$$Q_{21}(t) = \frac{\mu}{\mu + \lambda} [1 - e^{-(\mu + \lambda)t}] \quad (1.2)$$

$$Q_{24}(t) = \frac{\lambda}{\mu + \lambda} [1 - e^{-(\mu + \lambda)t}] ; \quad (1.3)$$

$$Q_{(M+4)(M+3)} = \frac{2\mu}{2\mu + \lambda} [1 - e^{-(2\mu + \lambda)t}] \quad (1.4) \quad (1.5)$$

* Shared Use Computer System

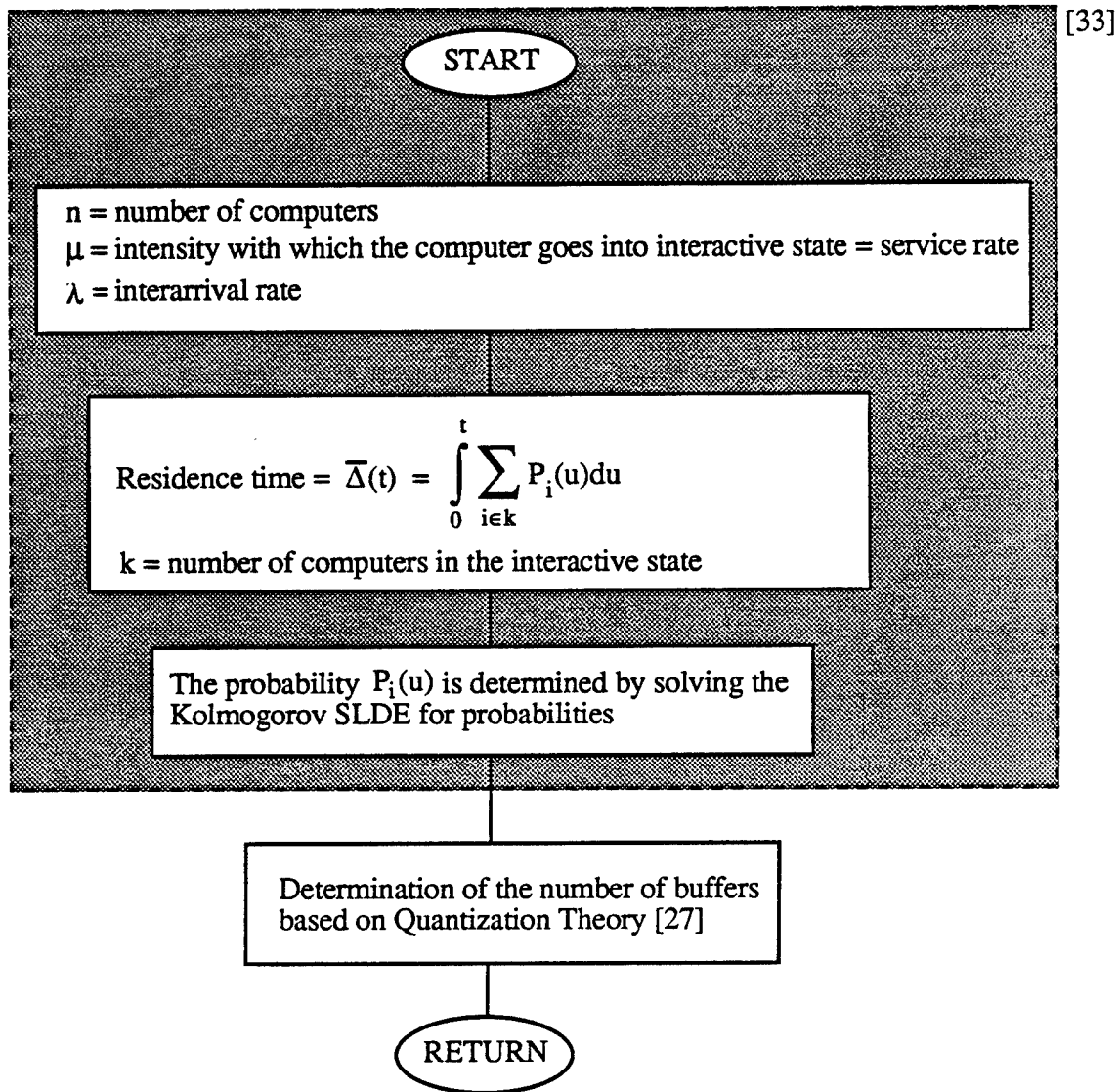


Figure 6. Residence Time Subalgorithm for m-Node Network Fragments

START

[25,43]

λ = interarrival rate into the buffer
 M = length of the buffer
 μ = service rate of all nodes ⁽²⁾ except central node
 η = service rate for central node

Mean stationary failure Prob.

$$G = C \left[\frac{\lambda^2}{(\mu + \lambda)(\mu + \eta + \lambda)} \right]$$

Mean residence time

$$\bar{q} = C \left\{ \frac{\mu(\mu\lambda + \mu + \lambda^2) + 2\lambda\mu^2}{\lambda^{M+2}(\mu + \lambda)(\mu^2 - \lambda^2)} (\mu^{M+1} - \lambda^{M+1}) + \frac{\mu^{M+2} - \lambda^{M+2}}{\lambda^M[(\mu^2 - \lambda^2)(\mu + \lambda)]} - \frac{(M+1)\lambda(\eta + 2\lambda)}{[\mu^2 - \lambda^2][\mu + \eta + \lambda]} \right\}$$

Mean failure free operating time

$$\bar{\tau} = \frac{\bar{q}}{G}$$

$$\bar{\tau} = \frac{\mu + \eta + \lambda}{\lambda^{M+4}(\mu - \lambda)(\mu + \lambda)} \left\{ (\mu^{M+1} - \lambda^{M+1})[\mu(\mu\lambda + \mu^2 + \lambda^2) + (\mu^{M+2} - \lambda^{M+2})\lambda^2(\mu + \lambda)] + \frac{M+1}{\lambda(\mu - \lambda)}(\eta + 2\lambda) \right\}$$

Calculation of the number of buffer on the basis of Quantization Theory [27]

RETURN

NOTE: Assumption (1) there are 3 computers, connected in a star shape
 (2) central computer has higher service rate.

Figure 7. Global Residence Time in a Star Network

$$Q_{(2M+8)(2M+7)} = \frac{2\mu + n}{2\mu + \lambda + \eta} [1 - e^{-(2\mu + \eta + \lambda)t}] \quad (1.6)$$

$$P_{ij}^* = \lim_{\eta \rightarrow \infty} Q_{ij} \quad (1.7)$$

Since the service rate of one computer cannot be approximated to be ∞ , the one computer in a LAN which is usually the central computer (gateway) can be approximated to have an infinitely high service rate. This model then enables the computation (Refer to Figure 7) of the star topology failure free operating time, and residence time considering the central node to have the higher service time.

If the messages are being routed in the ring, then we could use the Chou and Nilsson Algorithm [25] to find the delay, i.e. the residence time, associated with messages. Each node in the ring has two queues, one for locally transmitted messages and other for transit messages. The residence time calculation, as shown in Figure 8 for the ring topology, uses the interarrival rate, the service rate and the routing matrix. This delay is also dependent on the type of service protocol being used.

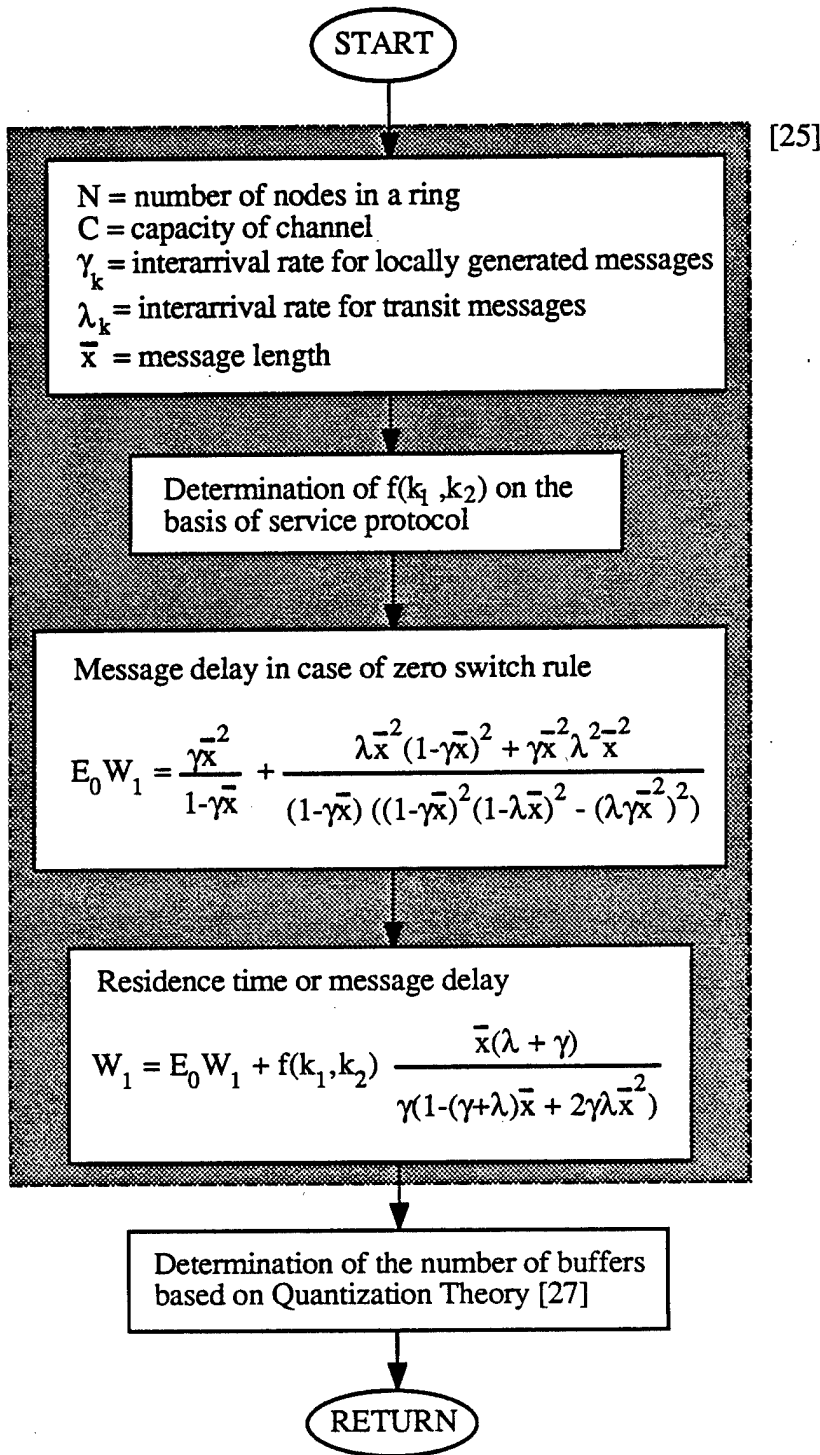


Figure 8. Calculation of Residence Time in a Ring Topology

2. Buffer's Availability And Grouping Or Sequencing Arrangement On The Basis Of Residence Time

Residence time, in a network fragment, consists of the summation of component times spent in the nodes and channels of the fragment. Some percentage of the residence time is spent in the nodes and some in the channels. The time spent in a node-channel pair can be determined from the number of nodes active in the fragment. This time can be further divided into the following three parts [27], (a) buffer delay + communication processor delay, (b) channel delay, and (c) accumulative node delay. From the buffer delay the mean waiting time for a message at position m in the buffer can be determined. The probabilistically weighted waiting time for a message at position m in the buffer can be determined as in Niznik [27]. From this parameter, the number of buffers occupied and free during the mean waiting time can be predicted if the probability of occupancy of these buffers is known from the sampled service interarrival rates.

3. Real Time Criteria Aspect For Simultaneous Inputs

Referring to Figure 3A, a control algorithm residing in the front end processor could be used to route the incoming messages to one of M input buffers. The control algorithm would not be required if there were one dedicated buffer to one input line. All M inputs now compete for access to the transfer processors. If some of these inputs lines are heavily loaded, i.e. high message arrival rate λ , this rate will give priorities to heavily loaded lines so they are processed first. However, a less loaded line will starve. Also, the FIFO algorithm could be used with the Dijkstra's software construct, the Semaphor*. In this software construct, messages competing for the same processor are placed in a queue, depending on their arrival, and their queue position is dependent on the value of the semaphor. There would be one global queue for all processors, which is updated every cycle. Then, processor 1 takes the first message in the queue, processor 2 takes the second message, etc. resulting in all the processors executing the messages

* Semaphor is a variable, that is modified by one process but cannot be modified by another process simultaneously.

simultaneously. At the completion of execution, the next available message in the queue is processed. This procedure ensures the prevention of starvation but the heavily loaded lines will wait and then other inputs lines will have increased waiting time resulting in a general degradation of throughput. To avoid degradation of throughput, the Round-Robin Scheme could be used. If the combination of FIFO and a priority algorithm is used, the links take turns in each time period, instead of one line having total access to the processor.

4. Multiple Access Aspects Of Contention Resolution. A Timing Cycle Solution To Contention

With multiple access to various processors, the contention resolution algorithm [24] is applicable where (1) the channel could be free, (2) there could be a successful transmission if there are only as many messages as processors, or (3) there could be contention of the number of messages when there are more messages (more input buffers) than the number of transfer processors. In this algorithm, after contention is observed, no new link can be active during the contention period. The contention period is divided into cycles of action for each message:

- (a) In the zero cycle some lines try to transmit. If the number of links is equal to the number of processors, then the attempt is successful. Otherwise, there is contention and cycle one is attained. In the real time system, the system clock can be used to implement the FCFS scheme with a variable clock which records the instance of arrival. The variable clock is used instead of the Dijkstra Construct to implement FCFS.
- (b) Prior to cycle one, each station generated a random binary number, which is uniformly distributed on the $(2^k - 1)$ interval, where k = number of stations for a sequential protocol grouping of messages in the buffer and the number of messages, when the inputs are not simultaneous.

- (c) At each successive cycle, a bit check is performed at startup from the m.s.b. (most significant bit)
- (d) At the end of the contention period there will be only n links left which could be less than the number of processors P .
- (e) If $n < P$ then there is no problem. However, if there is more than one link then a new contention period begins only for those links which have the first contention period left.
- (f) This step is a combination of the variable clock principle to implement FCFS following (e). If $n < P$ then the instant of arrival of each message is used to implement the FCFS scheme. Refer to Figures 9 and 15 for buffer arrangement and resolution.

When multiple access contention resolution is considered for the entire network, then all nodes are considered to be executing in parallel with multiple paths being executed simultaneously. If the same node happens to be in more than two paths and the messages in these paths reach the node simultaneously, then there is contention. In a star system like a satellite node, the problem of contention is more significant.

5. Calculation Of The Number Of Requests

The contention problem results from several messages being routed simultaneously when the number of messages (same as the number of input buffers) is greater than the number of transfer processors. To approximate the number of requests (messages) coming into a node, especially if a node has been temporarily disabled, as in a battle situation, the derivation is performed here in terms of two types of topologies, i.e. rings and stars. Since these specific computations are not real time, a real time approximation can be made if these requests are distributed linearly over the time period during which contention is considered.

(a) Calculation of the Number of Requests (Messages) For A Ring Topology

N processors can send data blocks of fixed length (requests) to one another for processing. Each processor can buffer up to N-1 requests. Each processor then addresses other processors equiprobably. A processor which has sent the request waits for the response before sending another. The servicing discipline is FCFS. Then, the probability of any processor receiving a request approaches 1 in the case of a heavily loaded network, and approaches 0 in case of a less loaded network, i.e. u is approximately 1 when the network Critical Applied Load is reached. The probability of a request being delivered to a link is r . r can approach 0 in the case of a heavily loaded network and 1 in the case of a less loaded network.

$$r \propto \frac{1}{\text{LOAD}}$$

(1.8)

Then, given the parameters r and u , the number of requests in a ring are modeled in the [48] equation n appearing in Figure 10. For the Critical Applied Load attainment the probability function is .99 and for no load .01. Then, the load can be calculated as a percentage of CAL and then u and r can be approximated.

6. The Load In A Real Time Computer Network

In Levin's paper [20,21], the job arrives at the computer when each of these jobs has m phase operating cycles. Phase 1 is the input, phase 2,..., $m-1$ are for data conversion and other processes and phase m is for output. These phases are in a pipeline. In a real time computer system, the instant of job arrival must be determined for the operating characteristics to be calculated, i.e. Average load R_i of phase i over time T . The information about the load must be used for two purposes: the load must be used in routing or congestion control software as a criteria to select or reject a path. If the

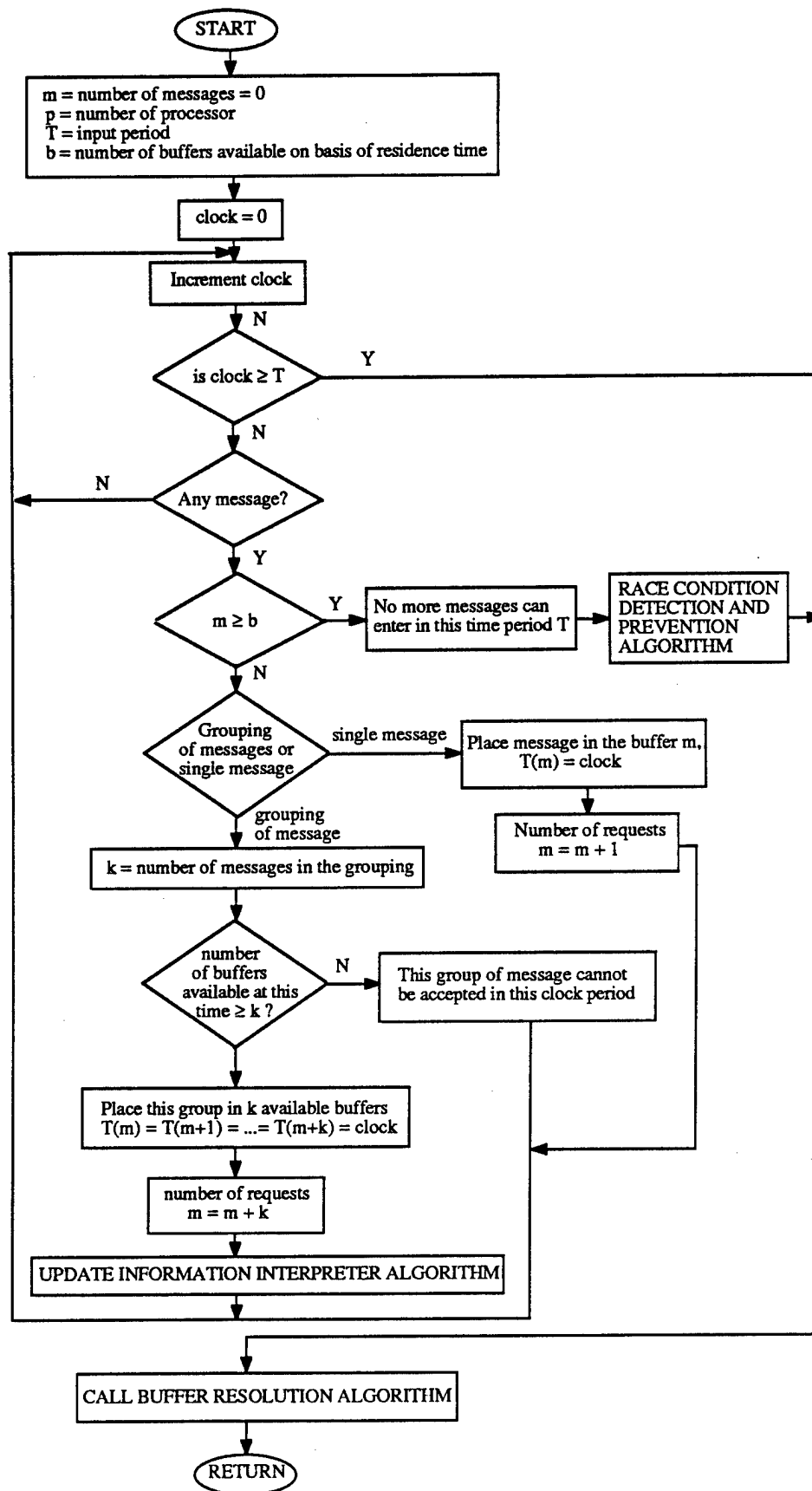


Figure 9. Real-Time Parallel and Sequential Input Buffer Arrangement Algorithm

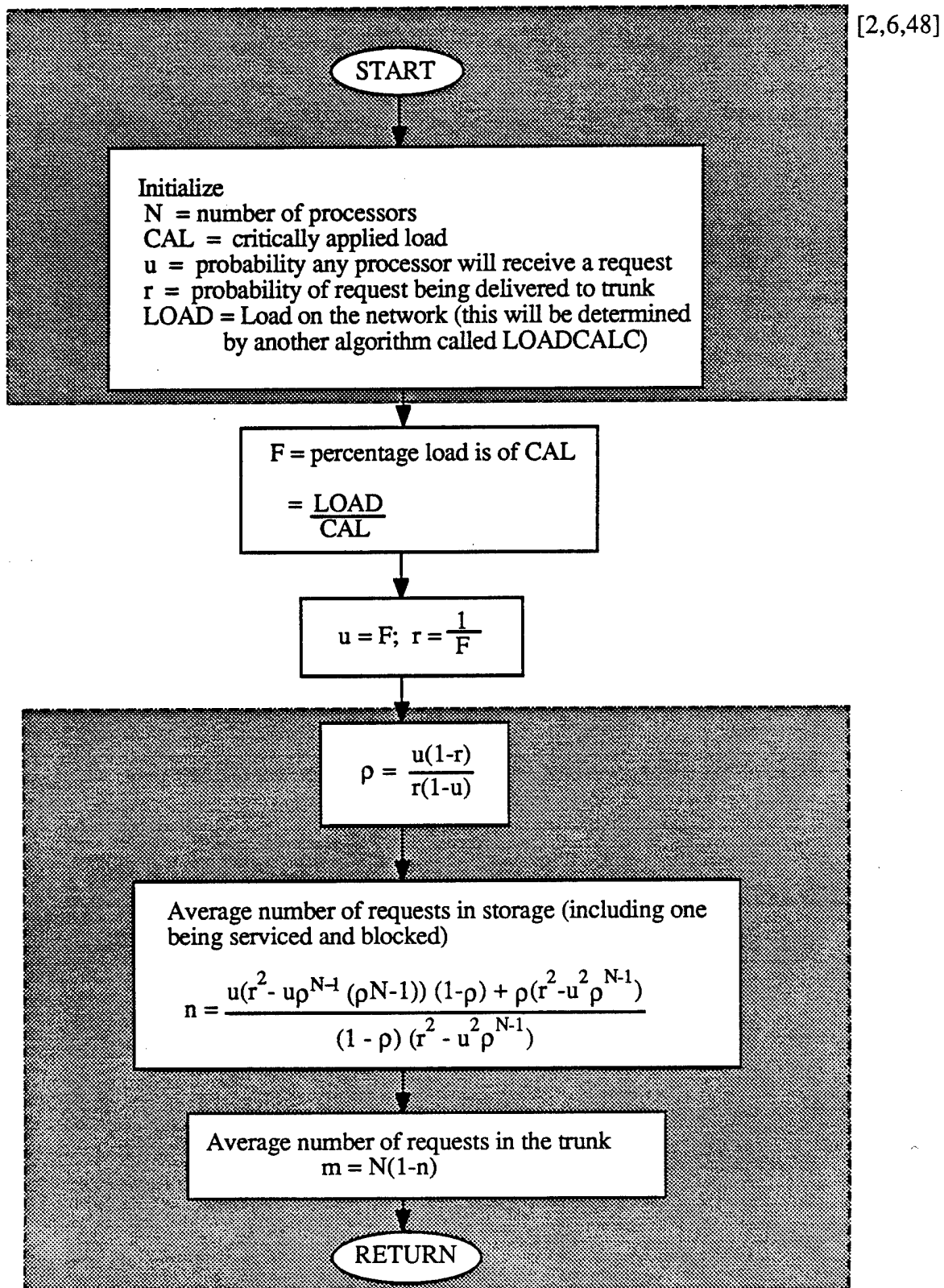


Figure 10. Number of Requests in a Ring Network Subalgorithm

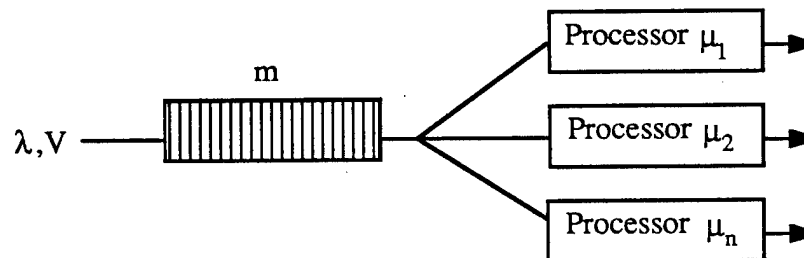
path load is greater than the Critical Applied Load the path is rejected unless it is selected for other, more compelling reasons. The load is also used for determining the number of requests in a ring network.

If all messages arrive in packets of different sizes, i.e., $P_1, P_2, \dots, P_p$, the processing time for each message at each phase is going to be the same, i.e. if at Phase 1 all messages of the same packet size take time t_{1P_α} ; at phase 2, all messages will take time t_{2P_α} . In a real time system using a FCFS, processing time of messages at the node is based on the interarrival rate λ . The matrix representing this load related processing time model is stated as:

$$A = \begin{matrix} & \begin{matrix} \text{Phase 1} \\ \text{Phase 2} \\ \text{Phase 3} \end{matrix} & \begin{bmatrix} t_{1P_\alpha} & t_{2P_\alpha} & t_{3P_\alpha} & t_{4P_\alpha} & \dots & t_{mP_\alpha} \\ t_{1P_\beta} & t_{2P_\beta} & t_{3P_\beta} & t_{4P_\beta} & \dots & t_{mP_\beta} \\ t_{1P_\gamma} & t_{2P_\gamma} & t_{3P_\gamma} & t_{4P_\gamma} & \dots & t_{mP_\gamma} \\ t_{1P_\delta} & t_{2P_\delta} & t_{3P_\delta} & t_{4P_\delta} & \dots & t_{mP_\delta} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ t_{1P_\eta} & t_{2P_\eta} & t_{3P_\eta} & t_{4P_\eta} & \dots & t_{mP_\eta} \end{bmatrix} \end{matrix} \quad \begin{matrix} (m+1) \times n \\ (1.9) \end{matrix}$$

This matrix will be used in the calculation of the load, where R = the average loading of the system for one processor. For multiple processors in a node, say k , the loads at that node will be divided by k for the load to be shared. The figures for the calculation of load are shown in Figures 11, 12, and 13.

7. Computation Of The Number of Requests In A Multiple Processor System



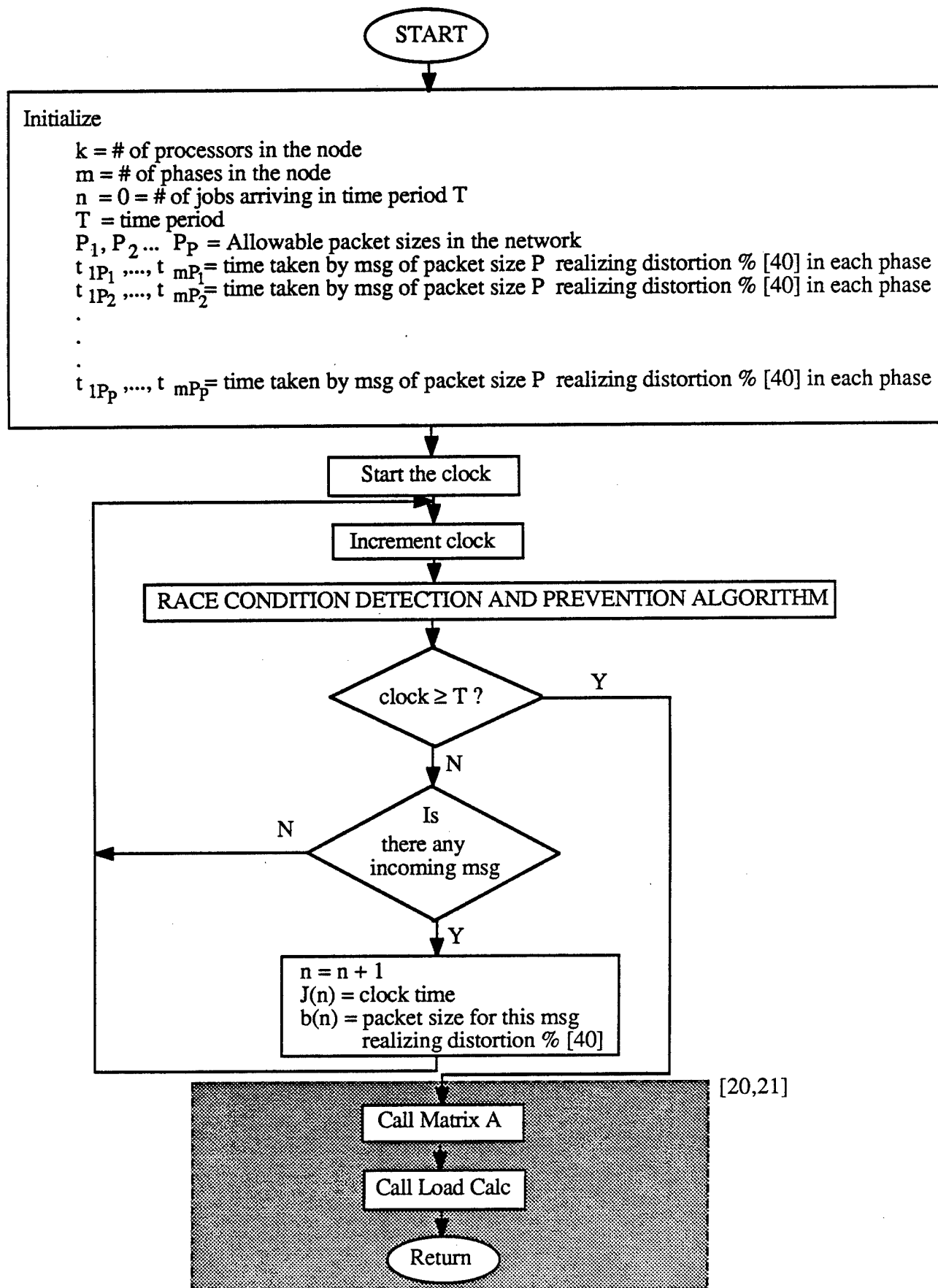
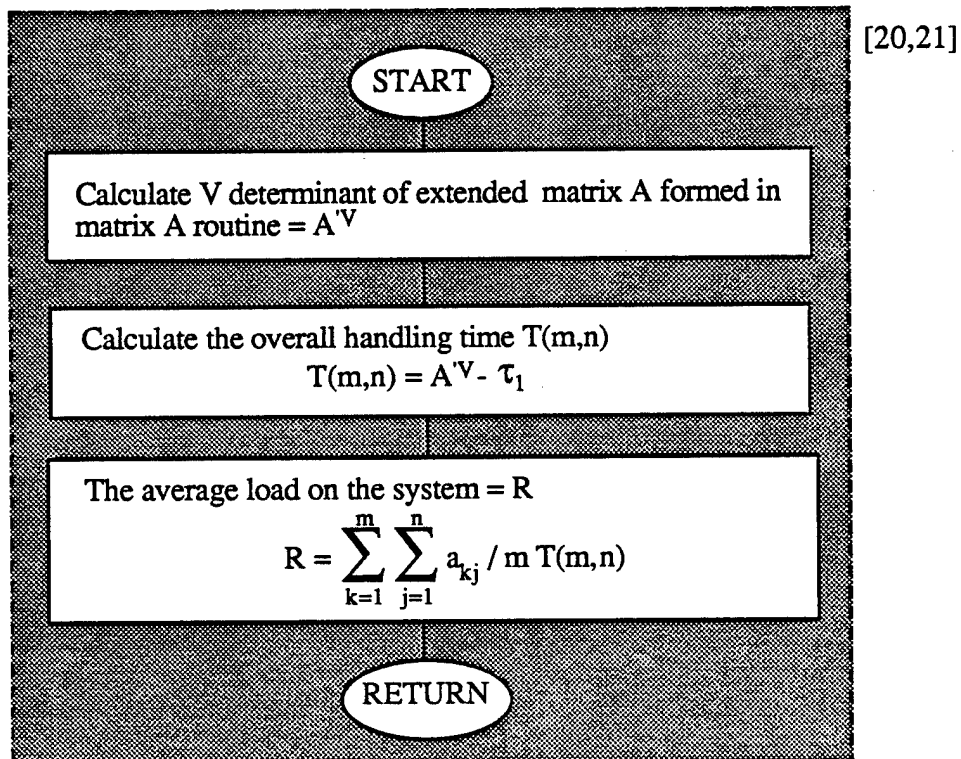


Figure 11. Load Calculation Subalgorithm at Each Node in the Network



- Note: (a) To be appended to the flowcharts after Matrix A.
 (b) The algorithm for the V determinant has not yet been written. This is the wave algorithm. The element of matrix A_{ij}^V are equal to the maximum element to the left and above them, added to the right element a_{ij} of matrix A.

Figure 12. Load Calculation Subalgorithm

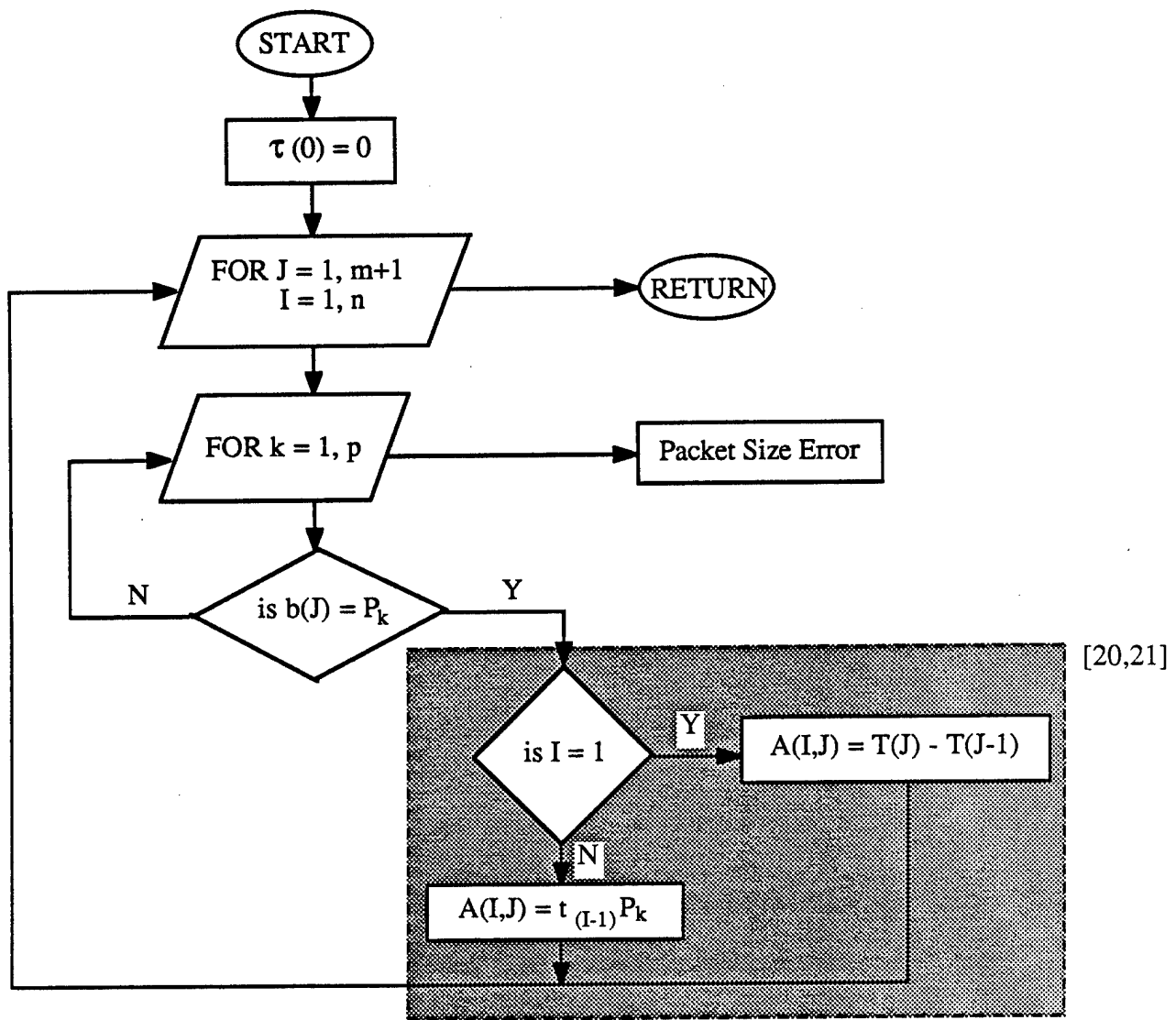


Figure 13. Matrix A

MODEL OF THE MULTIPROCESSOR SYSTEM

The previous figure describes a multiple computer system. Some of the nodes in the SDI satellite networks will consist of only one computer. $x(t)$ is defined here as a random process on a discrete set of states $(0, 1, \dots, m)$, where m is the maximum queue length, μ_i is the service intensity of a processor i , and there are n processors. The time over which the process $x(t)$ goes from state i to state $i+1$, $i=0, \dots, m-1$ is determined by the interarrival rate λ and variance v . If the process [22] $x(t)$ is described by a diffusion process $Y(x, t)$ with drift and diffusion process parameters $B(x)$ and $\alpha(x)$ respectively, the probability of the process $x(t)$ can be defined as π_i , which enables the solution of the number of requests coming into the buffer in a multiprocessor computer system. In this model, the number of requests either in the case of a ring or multiple processor system are modeled theoretically due to the possible failure mode of a node.

8. Calculation Of The Number Of Requests At A Node For A Star Topology

The star configuration [13,22] consists of a central computer as the upper level hierarchical node and N nodes in the lower level attached to the upper level node. The queueing system at both the upper the lower level receives a Poisson stream of requests with the interarrival rate and variance coefficient C_s . Requests are serviced in a FCFS manner. The probability of a request leaving the network is Q_k , a function of the service rate μ_k . On the basis of this descriptive model the distribution of the number of requests in the queue at each node in the network can be obtained. Then, the expressions for the steady state probabilities can be determined from the diffusion approximation or the Quantitization [27] technique. The number of requests entering the buffer can then be determined from these steady state probabilities as shown in Figure 14.

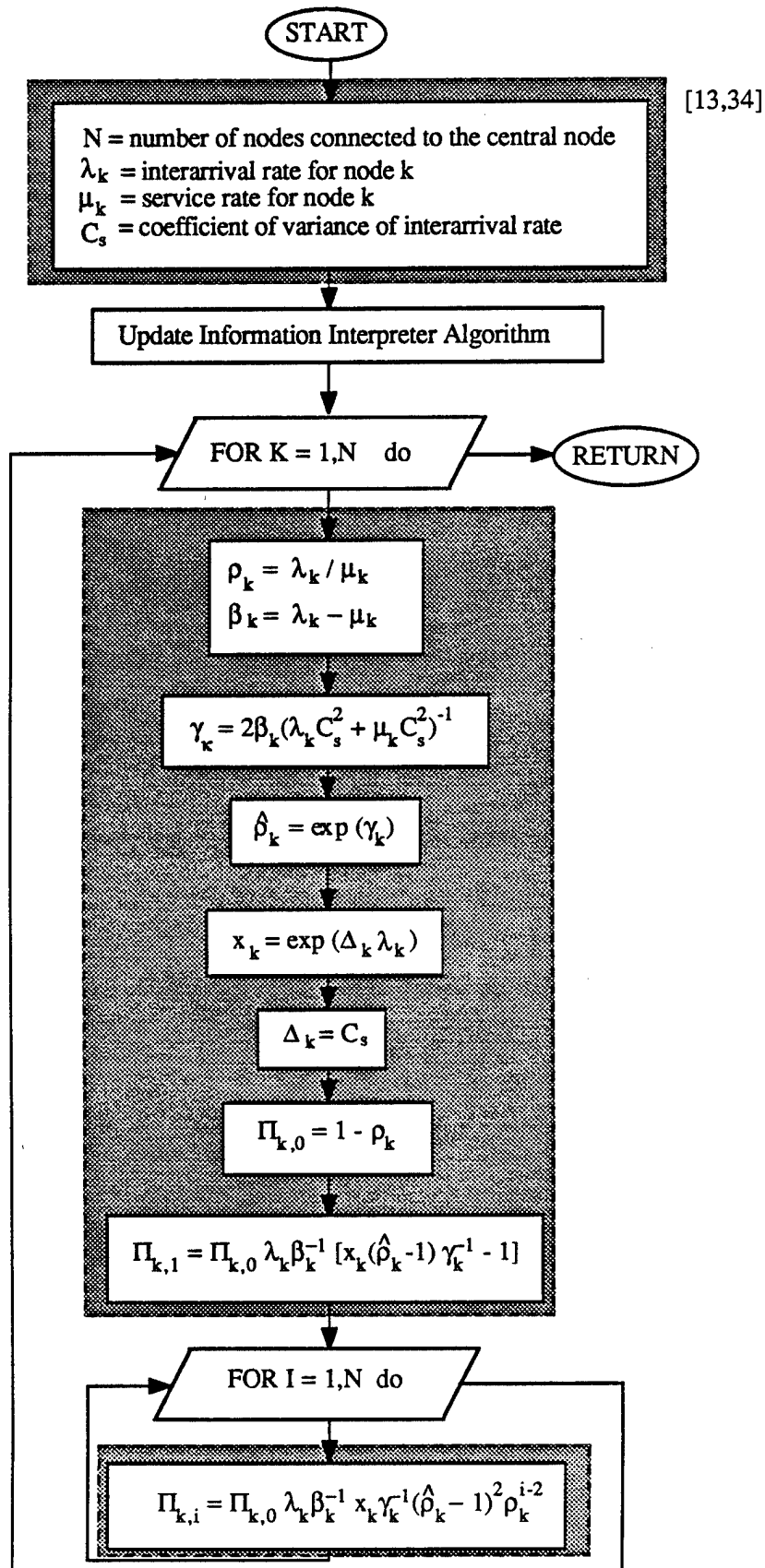
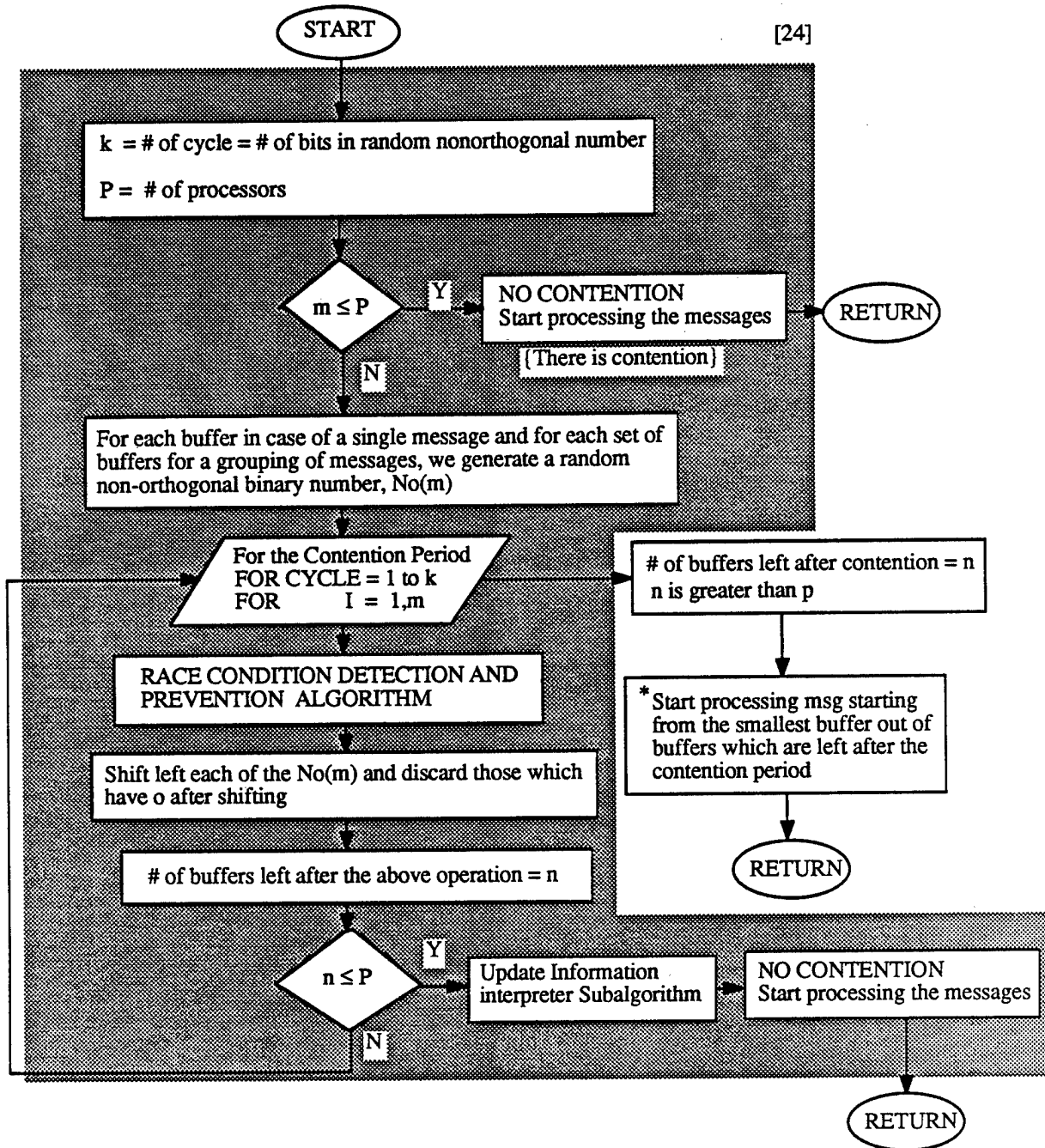


Figure 14. Number of Requests at all Nodes in a Star Topology



* Note: The reference [7,22] contention resolution algorithm above automatically places the messages starting from the smallest buffer number to the highest buffer number. So this automatically implements FCFS without going through Dijkstra or the Star Collision Algorithm [24] .

Figure 15. Buffer Resolution Subalgorithm

E. Real-Time, Parallel Synchronous and Asynchronous Inputs For Load Networks

1. Overview

The sequential logic design theory by G. A. Maley and J. Earl [23], references the following solutions to digital logic design circuit race conditions which can be incorporated in the CCSCR subalgorithm designated in the following manner. The three areas of computer networking problem generation: (1) real-time, (2) synchronous and asynchronous inputs, and (3) high load, are linked together and realized in the following two CCSCR subalgorithms (A) Real-Time Parallel and Sequential Input Buffer Arrangement Subalgorithm and (B) Load Calculation Subalgorithm At Each Node in the Network. The synchronous and asynchronous operations define the type of critical and noncritical races which can occur in the computer network. The synchronous mode of operation requires time allotment. Because of the synchronous mode's use of specific delay times, this operation of computer nodes occurs at a fraction of their potential speed. The most economical functional blocks of software, requiring the same response, regardless of input conditions, are produced by complex factoring, which results in functional blocks with different message path lengths for different input conditions. This will enable the development of a software function that requires a varying amount of time to complete its operation. An alternate solution is to reduce all network paths through a functional block to a minimum number of hierarchical levels. This solution increases in cost when the path length is decreased. In either of these solutions to synchronous network problems, there is the difficulty in predicting the exact time delay of the message packets, i.e. signals, passing through a series of logic connectives since no two computer nodes have identical characteristics and are alterable in time.

The synchronous mode is initiated by a signal (packet) from a complete signal (message) from the previous transmission. In asynchronous operation, only enough time is required to complete its defined message function. One of the concerns in solutions to optimal asynchronous computer node operation, is the overall average speed of the network function being designed. The asynchronous mode theory is that functional software that can be designed to

support message transmission when the input messages connected to the network function have been processed. To construct software for completely asynchronous operation requires twice the size of software code, computational storage, and time allocation as the synchronous operation.

2. Critical and Noncritical Races in Sequential Network Operations

To provide an operating system and front end processor sequential operation that enables parallel simultaneous operation, race conditions must be eliminated. Critical and noncritical race conditions will prevent simultaneous acceptance by the computer node of synchronous and asynchronous functional messages. In logic design sequential circuits, this type of problem, designated as eccentric, is dealt with in the excitation and transition matrices to observe the instability of the network. If there are two feedback message loops in the network that change a race condition, the net network state is defined by which feedback loop wins or ties the race. A noncritical race, a fixed race, is a race where the final operating state is insured. A critical race occurs when there is a multiple transition occurrence, and the network exists in either more than one non-stable state before reaching a stable operating state, i.e., a closed loop of non-stable states. In a logic design circuit the critical race condition indicates a network oscillating condition which can also be analogized to the computer and gateway satellite network operation.

3. Hazards In Sequential Network Operation

Hazards result from distinct differences in delays between paths propagating the same message/packet. The four types of hazards, three of which occur in the computer node when one variable parameter is changing, can be detected and corrected, two by changing the logic concepts and the third by delay insertion (or deletion), i.e. the static hazard, the dynamic hazard and the essential hazard. The fourth hazard occurs when more than one variable network parameter is changing and is a multi-variable hazard. The multi-variable can be any of the first types and can be dealt with by delay insertion or deletion. The static hazard and the dynamic hazard both are caused by

different path lengths of a message/packet in reaching the feedback, or ring message/packet loop. The essential hazard is observed in different path lengths for a message occurring in two different message/packet network ring topologies. The static hazard results in an output message changing transiently when it was supposed to remain static at one value during the alteration of a single network variable. The dynamic hazard results in an output message changing three or more times when it is supposed to change only once during the change of a network variable. The dynamic hazard varies from the static hazard since there must be three or more changing delay paths of a message to a single feedback situation. The essential hazard results in a false transition to an incorrect state of a sequentially operated computer node by changing the delays of a message to different message feedback responses. The essential hazard, which occurs only in sequential computer nodes, is caused by changing delays of a signal to different message feedback responses, such as packet triggering. Multi-variable static, dynamic and essential hazards contain the same conditions and several networking variables instead of a single changing variable.

In general, these hazards can not be completely eliminated and this is the reason that asynchronous computer nodes try to change only one network performance variable during a time interval.

4. Solutions to the Critical and Noncritical Race Conditions in Networks

The following techniques are included here within the Race Condition Detection and Prevention Algorithm (refer to Figure 16) in a computer networking congestion controlled software controlling manner for the CCSCR Algorithm:

- (a) Identification of equivalent states to enable mergers to minimize overall network state operation.

Referring to previous work in this equivalencing and reduction of states pp. 88, 89 in Niznik [28], such merging techniques of equivalent network states as the "multiple merger choice" rules are used in the CCSCR Race Condition Detection and Prevention Algorithm.

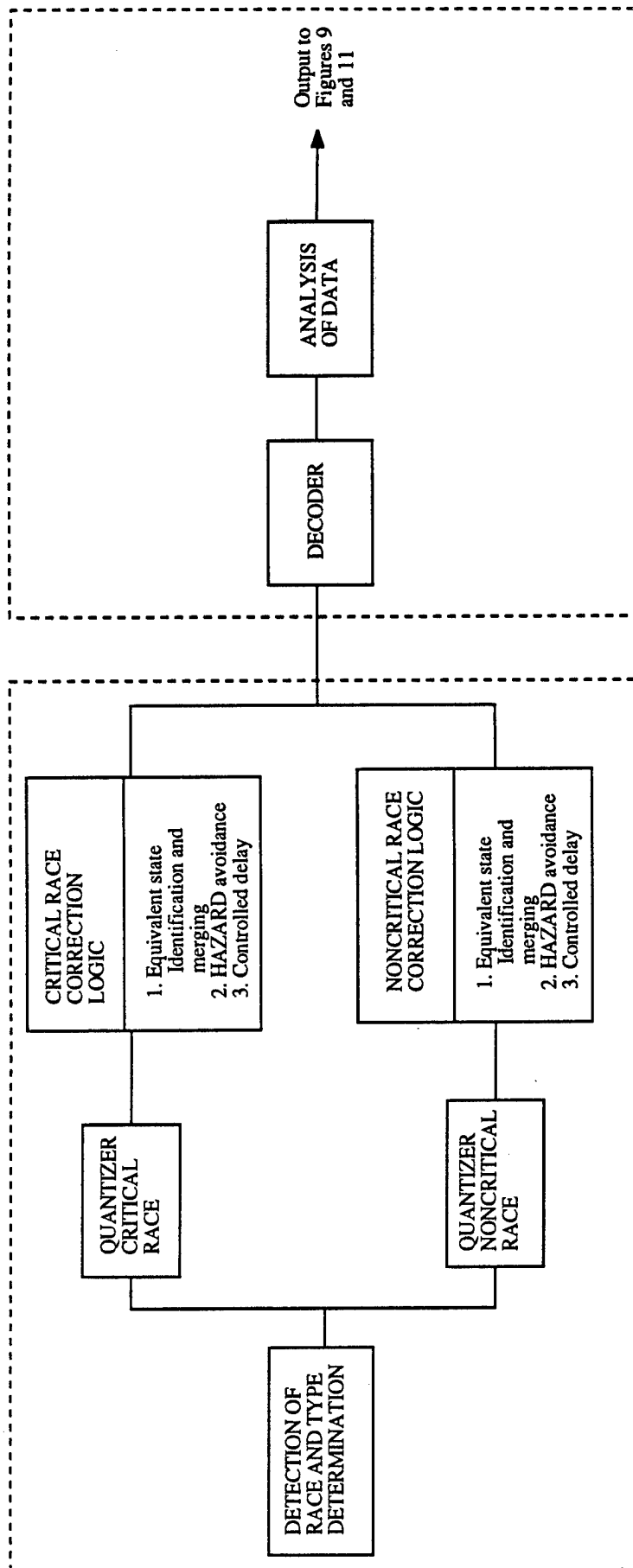


FIGURE 16: RACE CONDITION DETECTION AND PREVENTION ALGORITHM

(b) Elimination of Hazards.

All hazards in networks, circuit and computer, arise from specific differences in delays among paths propagating the same messages. Referring to the sequential circuit hazard theory by D. A. Huffman [12], the effects of removing these hazards is explored. Essential hazards occur only in sequential circuits and all hazards are sequential network problems. Therefore, essential hazards and their correlated networking instability, can be eliminated by inserting delays to diminish the likelihood of malfunctions due to the hazard.

(c) Delay Dependent and Delay Independent Operations.

Controlled delays which is a property of congestion and flow control techniques in computer networks, is a powerful tool, where critical races and deliberate software theory design errors may be used to significantly improve networking speed, reliability or cost.

4.0 PROBLEM STATEMENT: MESSAGE PATH LOAD AND PROPAGATION DELAY MONITORING IMPACT ON SEQUENCING SUBALGORITHM CONCEPTS

The addressing of survivability detection with respect to monitoring message path load and delay for imbedded security and privacy encoding using a cubic delay and load table software structure will be developed. Large path delays and the correlated path load impact will monitor track calculation errors.

PROBLEM SOLUTION

Abstract

The most significant relationship in the CCSCR Algorithm's sequencing concepts related to path load and propagation delay monitoring, is their connection mathematically to the transmission packet block size. The packet size intuitively has a direct link to distortion of the packetized message. The packet size should be increased to reduce the proportion of overhead bits in the packets and to reduce the relative time that the channel is not busy. However, increasing the packet size entails an increase in the per packet error level and reduction in the number of buffers for packet storage capacity, thus resulting in an increase in the number of retransmissions.

As the path load increases, i.e. more computer and gateway nodes compete for the same channel, then the path delay also increases. An increase in the path load indicates that more messages are waiting to be transmitted. These messages can be destroyed or changed by outside factors, i.e. protocol jamming. The following are methods to solve the problem of message block distortion:

(A) Multiple Access Solution

Instead of sending the messages on a single channel, they can be transmitted simultaneously to the neighboring nodes and the routing protocol will ensure that the messages reach the final destination. In this case, if the message

is jammed in one channel, then other channels in the network will transmit the message to its destination.

(B) Survivability Solution

The second technique is to break the messages into sections which may be packets of different size. In this case jamming will affect only a limited number of packets, and only that percentage of packets will require retransmission. This optimal packet size and the resulting percentage of packets requiring retransmission, will allow a specific level of distortion service.

(C) Security and Privacy Solution

The third technique is to encode the message within the cubic delay and load table and/or in the other sections of network transit in such a manner that the message becomes indestructable, i.e., unaffected by protocol jamming.

IV. INTRODUCTION AND OVERVIEW

In order to link the network path delay and path load to packet (block) distortion, a Lagrangian formulation is used in connecting delay and load individually or in a multiple manner as constraints to the distortion probability of message bits. This optimization of the message bit distortion is derived by differentiating the probability of bit distortion realizing a throughput maximization. To obtain the Lagrange multipliers under the allowance of a specific packet distortion percentage in the objective function, the constraints of maximum load % and maximum congestion % are used. A multicriterion optimization equation (2.1a) or single criterion optimization equation (2.1b, 2.1c) are realized in the CSSCR software to develop the optimized packet bit distortion percentage r_{opt} .

$$r_{opt} = r_{obj} + \alpha [CAL - k] + \beta [K - \zeta] \quad (2.1.a)$$

$$r_{opt} = r_{obj} + \alpha [CAL - k] \quad (2.1.b)$$

$$r_{opt} = r_{obj} + \beta [K - \zeta] \quad (2.1.c)$$

where r_{opt} = packet bit distortion %
 r_{opj} = packet bit distortion percentage (prior to optimization)
 CAL = Critical Applied Load
 k = Load %
 α = Lagrange multiplier for load %
 K = congestion allowed
 ζ = congestion percentage
 β = Lagrange multiplier for congestion %

The path delay is the summation of queueing buffer delays, propagation delays, nodal delays and channel delays. The derivative for the solution of the Lagrange multiplier will require differentiation with respect to the packet distortion constraint in the following derivation.

1. Linking of Message Distortion to Contention and Congestion %

The higher contention and congestion level of service enables a higher message distortion. If congestion is allowed in the SDI network, then, thrashing the return to the source node for retransmission will occur for packets which are lost or distorted. In the case of a heavily congested computer network, the operation of returning incomplete or distorted packets is the definition of severe thrashing. The packet size is related to distortion in the packets by the following mathematical formalism alluded to by Sushchenko [39).

2 Packet Size Determination.

The length of a packet is chosen in such a manner by Sushchenko [39] that throughput can be maximized and message delays decreased for a fragment of a packet-switched network realizing channel characteristics, protocol parameters, and blocking of storage of the receiving node. When a packet is transmitted from one node to another node in the network, then there are two transmission possibilities:

- (a) The packet is transmitted successfully, and
- (b) There is some distortion in the packet due to protocol jamming. In this case the packet has to be transmitted again.

Packet size is usually a multiple of 2. The probability of distortion increases if the congestion in the link or the contention at the receiving node increases. This statement assumes that protocol jamming avoidance software is properly functioning. The packet size should be increased to reduce the proportion of overhead bits in packets, and to reduce the relative time that the channel is not busy. Increasing the packet size increases the per-packet error level and a reduction in the number of buffers for packet coverage for dedicated buffer storage capacity, thus resulting in an increase in the number of repeat transmissions. In order to utilize the message packet distortion as a constraint function for the path load or path delay, Suschenko's [39] packet load/distortion percentage derivation is restated and explained with respect to this task in equations (2.2) to (2.7) in the following.

The time of a total packet transmission cycle is,

$$t_m = mt + T_m, m = 1,2 \quad (2.2)$$

where $m = 1$ for unidirectional operation
 $m = 2$ for bidirectional operation

If R_1 and R_2 are the probability of packet distortion in the link of the first and second link of the message path respectively, and N is the number of buffers on the output channel, then, the distribution of state probability assuming $R_1 = R_2 = R$, are expressed by Q_i ,

$$Q_i = \begin{cases} R/(N+R), i = 0 \\ 1/(N+R), i = 1, N \end{cases} \quad (2.3)$$

Other conditions of interest are:

- I. Absolutely Reliable Channel of the Second Data Link ($R_2 = 0$), $Q_0 = R_1$,
 $Q_1 = 1 - R$, $Q_i = 0$, $i = 2, N$.
- II. Busy State $R_1 = 0$, $Q_1 = 1$,
- III. Deterministic servicing of packet flow - one buffer for storing the packet
in the queue for the output.

Assuming the principal contribution during repeat transmission comes from the first repeat transmission, the average packet transmission time is,

$$t_m = (mt + T_m) [(1+R + (1-R) Q_N)] \quad (2.4)$$

where

- $(1-R)Q_N$ = probability of repeat packet transmission
- Q_N = blocking probability as a result of the quality of the
received and service message flows
- $(\lambda(1-Q_N))$ = $\mu(1-Q_0)$
- λ = interarrival rate
- μ = service rate

Defining the throughput $C(L)$ as the throughput of the first data link whose units are bits/sec. And "disregarding quantities proportional to" RQ_N ,

$$C(L) \approx (L-H)/[(mt+T_m) (1+R + Q_N)] \quad (2.5)$$

where

- H = the number of overhead bits
- R = $1 - (1-r)^{KL} = rKL$ = probability of packet distortion
- r = independent probability of bit distortion
- $K \gg 1$ = the coefficient of increase of the packet length as a
result of bit (or byte) stuffing or use of asynchronous
communication devices.

For large buffer sizes $Q_N \approx 1/N$. If N is a continuous variable, $N = V/(L+h)$, where V = the buffer storage allocated to the output channel, and h is the number of bits in the buffer required to organize a queue in the structure of a coupled list of buffers. The packet size L obtained by optimizing the throughput, for $K = 1$, an unlimited transit mode buffer storage, and arbitrary window size is stated:

$$L = H \sqrt{[H + (ST_m/M)] (H + 1/r)}$$

where,

(2.6)

S = the physical transmission rate on the channel (bits/second)

Since packet size is usually in powers of 2, the following expression results for the optimal packet length for arbitrary window size,

$$L = H + \begin{cases} 2^n, L - H \leq 2^7 \\ 2^n, 2^n \leq L - H < 2^{k+1}, \nabla_n \geq 0, n = \overline{7,12} \\ 2^{n+1}, 2^n < L - H \leq 2^{k+1}, \nabla_n < 0, n = \overline{7,12} \\ 2^{13}, L - H \geq 2^{13} \end{cases}$$

where,

(2.7)

$$\Delta_n = C(2^n + H) - C(2^{n+1} + H)$$

Higher contention and congestion causes higher bit distortion. Here, the congestion % is linked to the packet and therefore message distortion.

3. Linking of Distortion to Contention and Congestion.

Rearranging equation (2.6) in terms of R in the probability of packet distortion, we have

$$(L-H)^2 = (H + (ST_m/m) (H+1/r))$$

$$L^2 - 2HL + H^2 = H^2 + \frac{HST_m}{m} + \frac{H}{r} + \frac{ST_m}{m} \left[\frac{1}{r} \right]$$

$$L^2 - 2HL - \frac{HST_m}{m} = \frac{1}{r} \left[H + \frac{ST_m}{m} \right]$$

$$\left[L^2 - 2HL - \frac{HST_m}{m} \right] = \frac{1}{r} \left[H + \frac{ST_m}{m} \right]$$

$$r = [H + (ST_m/m)] / [L^2 - 2HL - (SHT_m/m)]$$

(2.8)

Using the Lagrangians shown in equations (2.9a) and (2.9b), r , the bit error objective function, is optimized with respect to L , the packet length,

$$r_{opt}|_c = \left[[H - (ST_m/m)] / [L^2 - 2HL - (SHT_m/m)] \right] + \alpha [CAL - k] \quad (2.9a)$$

$$r_{opt}|_k = \left[[H - (ST_m/m)] / [L^2 - 2HL - (SHT_m/m)] \right] + \beta [K - \zeta] \quad (2.9b)$$

We know from A. Giessler, J. Hanle, A. Konig, E. Pade [9] that the Critical Applied Load in a network (CAL) is,

$$CAL = NVC (\min[(CAP/L)/NAVCL]) \quad (2.10a)$$

where,

NVC	=	number of virtual channels in the network
CAP	=	capacity (bits/sec)
L	=	packet length (bits)
NVCL	=	number of virtual channels in the network link

and from Niznik [26] the % of network congestion, K is stated,

$$K = \left\{ \max_{i=1} K_i \right\} = \max \left[\frac{\lambda_i \mu_i^{-1}}{C_i - \lambda_i \mu_i^{-1}} \right] = \frac{\text{average traffic}}{\text{excess available capacity}}$$

$$K = (\lambda \mu^{-1}) / (C - \lambda \mu^{-1}) \quad (2.10b)$$

where

n	=	number of i links
λ_i	=	average traffic along ith branch
C_i	=	capacity of link (bits/second)
K_j	=	link i congestion
μ^{-1}	=	independent exponentially distributed average message lengths (bits/mesg).

The following equations

- (1) substitute equations (2.10a) and (2.10b) into equations (2.9a) and (2.9b), respectively,
- (2) replace L by λL in equation (2.10a) and $\lambda L/g$ in equation (2.10b), where g = number of packets/msg, to enable insertion of the alteration %, for the packet length,
- (3) differentiate $r_{\text{opt}} \Big|_c$ with respect to L and equate it to zero to obtain the value of β .

Solving for the bit distortion $r_{\text{opt}}|_c$ with the load constraint,

$$\begin{aligned}
 r_{\text{opt}}|_c &= \frac{[H + (ST_m/m)]}{[L^2 - 2HL - (SHT_m/m)]} + \alpha[CAL - k] \\
 &= \frac{H + (ST_m/m)}{L^2 - 2HL - (SHT_m/m)} + \alpha \left[NVC \left(\frac{CAP}{\gamma L (NVCL)} \right) - k \right] \\
 &= \frac{H + (ST_m/m)}{\gamma^2 L^2 - 2H\gamma L - (SHT_m/m)} + \alpha \left[NVC \left(\frac{CAP}{\gamma L (NVCL)} \right) - k \right] \\
 r_{\text{opt}}|_c &= \frac{H + (ST_m/m)}{\left[\gamma^2 L^2 - 2H\gamma L - \frac{SHT_m}{m} \right]} + \alpha \left[\left[\frac{NVC(CAP)}{\gamma L (NVCL)} \right] - k \right] \\
 \frac{\partial r_{\text{opt}}}{\partial L} \Big|_c &= (H + (ST_m/m)) \left[\frac{\partial}{\partial L} \left[\frac{1}{\gamma^2 L^2 - 2H\gamma L - (SHT_m/m)} \right] \right] + \frac{\partial}{\partial L} \left[\left[\frac{\alpha NVC(CAP)}{\gamma L (NVCL)} \right] - k \right] \\
 \frac{\partial r_{\text{opt}}}{\partial L} \Big|_c &= (H + (ST_m/m)) \left[-1(\gamma^2 L^2 - 2H\gamma L - (SHT_m/m))^{-2} (2\gamma^2 L - 2H\gamma) \right] - \left[\frac{\alpha (NVC)(CAP)}{(\gamma^2 L NVCL)^2} \right] = 0 \\
 \alpha &= \frac{(H + (ST_m/m))(2\gamma^2 L - 2H\gamma)(\gamma^2 L (NVCL))}{(\gamma^2 L^2 - 2H\gamma L - (SHT_m/m)) (NVC) (CAP)}
 \end{aligned}$$

(2.11a)

substituting α in equation (2.9a),

$$r_{\text{opt}} \Big|_c = \frac{H + (ST_m/m)}{\gamma^2 L^2 - 2H\gamma L - (SHT_m/m)} + \frac{(H + (ST_m/m)) (2\gamma^2 L - 2H\gamma)(\gamma^2 L)(NVCL)}{(\gamma^2 L^2 - 2H\gamma L - (SHT_m/m))^2 (NVC)(CAP)} [CAL - k] \quad (2.11b)$$

Then, solving for the bit distortion $r_{\text{opt } k}$ with the congestion % constraint,

$$\begin{aligned} r_{\text{opt}} \Big|_k &= r + \beta \left[(\lambda \mu^{-1}) / (C - \lambda \mu^{-1}) - \zeta \right] \\ r_{\text{opt}} \Big|_k &= \frac{H + (ST_m/m)}{\gamma^2 L^2 - 2H\gamma L - (SHT_m/m)} + \beta \left[(\lambda (\gamma L/g)^{-1}) / (C - \lambda (\gamma L/g)^{-1}) - \zeta \right] \\ \frac{\partial r_{\text{opt}}}{\partial L} \Big|_k &= - \left[(H + (ST_m/m)) (2\gamma^2 L - 2H\gamma) \right] / \left[\gamma^2 L^2 - 2H\gamma L - (SHT_m/m) \right] \\ &+ \beta \left[\lambda \left[\frac{- (C - \lambda (\gamma L/g)^{-1}) ((\gamma L/g)^{-2} (\gamma/g)) - [(\gamma L/g)^{-1} (\lambda (\gamma L/g)^{-2} (\gamma/g))]}{(C - \lambda (\gamma L/g)^{-1})^2} \right] - \zeta \right] = 0 \end{aligned}$$

Solving the Lagrange multiplier β ,

$$\beta = \frac{[(H + (ST_m/m)(2\gamma^2 L - 2H\gamma)]}{[\gamma^2 L^2 - 2H\gamma L - (SHT_m/m)]} \left/ \left[\lambda \left[\frac{-\left(C - \lambda(\gamma L/g)^{-1}\right)\left((\gamma L/g)^{-2}(\gamma g)\right) - \left[(\gamma L/g)^{-1}\left(\lambda(\gamma L/g)^{-2}(\gamma g)\right)\right]}{\left(C - \lambda(\gamma L/g)^{-1}\right)^2} \right] - \zeta \right] \right. \quad (2.12a)$$

Therefore,

$$r_{opt} \Big|_k = \left[\frac{H + (ST_m/m)}{\gamma^2 L^2 - 2H\gamma L - (SHT_m/m)} \right] + \beta \left[\lambda \mu^{-1} / (C - \lambda \mu^{-1}) \right] - \zeta \quad (2.12b)$$

The above two equations for bit distortion respectively, relate packet distortion to network load contention monitor (equation (2.11b)) and packet distortion to congestion and the interarrival and service rates (equation (2.12b)). Since congestion in the path usually determines the path delay, the distortion or the packet size L can be linked to the path delay and the path load. the overall impact of this packet distortion optimization with respect to the percentage of congestion or the percentage of contention load is reflected in the CCSCR Algorithm equations by the choice of an independent bit error r which gives an optimal approximation of the optimal packet length L_0 for q values that are commensurate with or less than r . Realizing that the cost of transmission is usually greater than the cost of data storage and processing, the buffer size V should be determined in a manner that q is at least amenable with the bit error level r .

Since r , the independent bit distortion is also related to the packet length distortion by $r = R/KL$, then equation (2.8) will become the following expression for the packet length distortion R ,

$$R = KL [H - (ST_m/m)] / [L^2 - 2HL - (SHT_m/m)] \quad (2.13)$$

Then, $R_{opt} \Big|_c$ and $R_{opt} \Big|_k$ would be defined:

$$R_{opt} \Big|_c = KL \left[\frac{H - (ST_m/m)}{L^2 - 2HL - (SHT_m/m)} \right] + \alpha [CAL - k] \quad (2.14)$$

$$R_{\text{opt}}|_k = KL \left[\frac{[H - (ST_m/m)]}{[L^2 - 2HL - (SHT_m/m)]} \right] + \beta [K - \zeta] \quad (2.15)$$

This macroscopic approach will yield the α and β for the optimization of the packet distortion %, R_{opt} .

5.0 PROBLEM STATEMENT UPDATE INFORMATION INTERPRETER SUBALGORITHM DEVELOPMENT

An input population monitor by a mathematical error code generation for the battle manager will enable (parity) execution of the definition of the critical route determination task of the battle manager and the optimal route selection task of the network manager. These error codes can be linked in time to the battle signature during the evaluation of the battle sequence in time to provide a parity snapshot of monitored input errors at various critical time intervals during the battle cycle. Identified battle sequences determined by the hashing technique employed will become part of a linked list in the software formalism for future reference.

PROBLEM SOLUTION

ABSTRACT

This task utilizes the following critical conceptual and theoretical issues of Sections 3 and 4 respectively to enable critical route determination and parity encoding: (1) race condition resolution algorithm, and (2) the determination of the distortion % for different packet block sizes correlated to contention resolution and congestion % allowance, under heavy load circumstances. The parity snapshot of input error requires a parity input coded sequence and the correlated time frame check word generation at the input to all SDI computer nodes and gateway computer nodes. The critical route selection software theory will realize specifically critical and noncritical race condition impacts on parallel input computer and gateway computer nodes, appended to the CCSCR congestion control techniques for various topologies. Packet triggering is linked here to critical and noncritical races, because these timing conditions will result in packet delivery at inconsistent times, especially for sky and ground networks interacting together. The packet distortion problem is also magnified because extremely incorrect message transmission will occur due to incomplete packets and the distribution of messages and their correlated packets.

V. INTRODUCTION TO HASHING TECHNIQUES

Hashing is a method in software or hardware for organizing large external data files such as the SDI Architecture track files, which require fast retrieval and retrieval speeds completely independent of the file size. Two recent state of the art techniques which are most relevant to the hashing technique required for the Update Information Interpreter Subalgorithm are dynamic hashing [17] and one-probe hashing [17]. The Dynamic hashing technique is capable of manipulating files, a collection of records with a primary key that grows and shrinks dynamically without degradation of retrieval performance, storage utilization and periodic reorganization. The One-probe hashing enables any record to be retrieved in disk access. These two techniques were implemented in the parallel hash table developed to implement this task. All hashing methods perform the following functions:

- (1) When the storage utilization increases, the retrieval time and the time of all other operations increases.
- (2) When the load of a file increases over a certain space and a local reorganization is completed. The rate of this process depends on the number of records stored in the file.

A. Update Information Interpreter (UII) Subalgorithm Theoretical Derivation

The UII Subalgorithm is generated from the CCSCR basis parallel simultaneous input requirement and is therefore a parallel hashing software algorithm composed of conceptually and theoretically the three major parallel hashing hardware components referenced by S.S. Thakkar and A. E. Knowles [41], a hash address generator, a hash table and a hash controller. Here, the parallel hashing techniques with the most minimal retrieval of databases, encompasses an optimization techniques to maintain the optimal sampling interval for data applied in a Least Square Distance curve fit. For the SDI Architecture requirement of the linkage of error codes in time, to the battle signature during the evaluation of the correlated battle sequence, the Gonnet and Larson [11] techniques is utilized here for increasing the database retrieval speed by an amount of internal storage with fixed signatures and separators. This

is essentially an extending (multiple) semaphore which shares multiple parallel process. A semaphor, a shared memory cell, is the definition of two parallel processes which rely on each other for their system operation and the triggering of other processes with a form of elementary communication between them to occur at significant time intervals.

The hash table is structured for this parallel apparently simultaneous input organization requirement, in z parallel sections of θ words. The entries, i.e. the inputs correlated to the battle sequence, from each bank are input in parallel using the hash address. The use of fixed length signatures and separators correlated to the battle signature in time, handled overflow records by open addressing without links or points. Therefore, an internally stored separator table is required which contains m separators each Ω bits in length. Also, a comparator between a theoretical key K and an accessed key $K_{\delta e}$, ($1 \leq \delta \leq D$, $1 \leq e \leq E$). The record with key K will have a correlated probe sequence $H(K) = (h_1(K), h_2(K), \dots, h_m(K))$. Therefore, the probe sequence is individually determined by the key and specifies the order that pages will be monitored for inserting or retrieving the record. The signature sequence, correlated to the battle signature, $S(K) = (s_1(K), s_2(K), \dots, s_m(K))$ contains a k bit integer so that it is individually determined by the key of the record. Thus, summarizing, when the record with key K probes page $h_i(K)$, signature $s_i(K)$ is used, where $i=1, \dots, m$. The separator table is used in conjunction with the battle signature where there is a page where, $r, r > b$ records hash. Since the page only can store b records, $r-b$ records will be lost after attempting its next probe sequence page. The r records are stored on their current signature. Signatures values separating the groups are placed in the separator table.

A partitioning is sought where the highest signature in the first grouping is different than the lowest signature in the second group. Therefore, using this technique, a record can be obtained for processing in one disk access, which is a critical system landmark requirement here realizing a parity bit monitoring availability of data in the hash table along with the battle signature security identification for parallel input data realizing a consistent parallel hashing single disk access technique. Refer to Figure 17

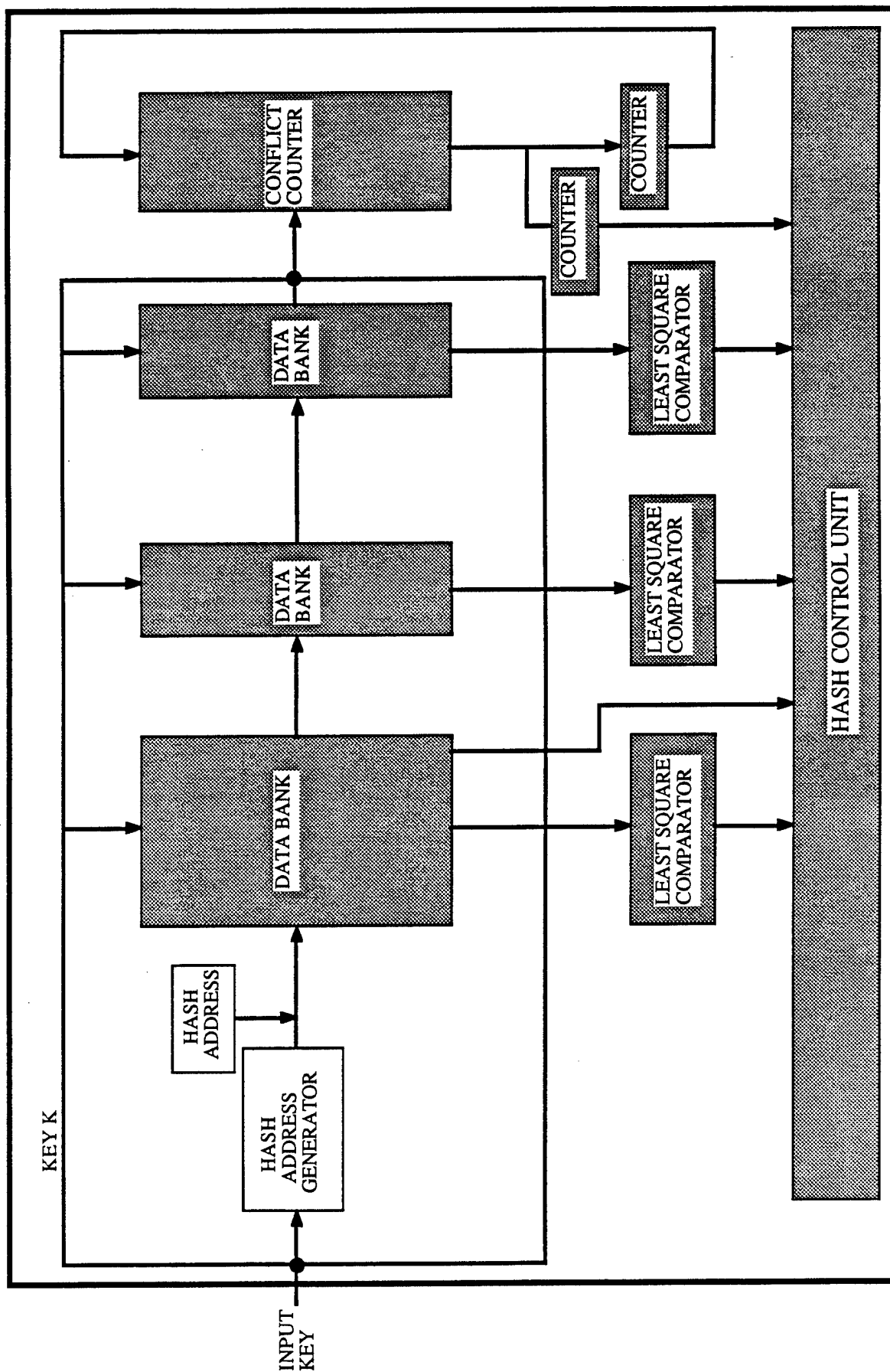


FIGURE 17. UPDATE INFORMATION INTERPRETER SUBALGORITHM

for the UII Flowchart and the following derivation for the UII comparator formalism which determines the optimal critical route determination. In this parallel hash table of z banks, the first entry is considered a primary location for data accessed by a hash address and other entries in the line are secondary locations (D,E entries). Then, the main location and the D-E secondary locations are input in parallel.

The parity execution of the critical route determination task of the battle manager and the optimal route selection task of the network manager can be achieved by the empty bit $Y_{\delta,e}$ associated with each entry when it is empty or occupied.

B. UII Parallel Hashing Software Load Factor

For the parallel hash software table in the UII subalgorithm, the load factor L_F is defined in by Thakkar and Knowles [41], as

$$L_F = N / (D \cdot E) \quad (3.1)$$

where,

N	=	number of real page frames or maximum number of active entries in the hash table	
D	=	number of banks	
$E-1$	=	number of secondary locations	
L_{Fm}	=	$N_m / (D \cdot E)$ = maximum load factor	(3.2)

Then, a least squares fit optimization is realized for the population data to obtain the optimal signature time sampling interval. The optimal route can be obtained with the following expression realizing also the population optimization during the imposition of the upper bound load factor of the hash table. To obtain the optimal sampling interval h for optimal routing considerations to be accurately complete, the following least squares expression comparing the actual population density distribution with the SDI Architecture theoretically predicted population, must be solved.

$$\left. \frac{\delta d(f,g)}{\delta h} \right|_{L_F} \left[\sqrt{\sum_{i=1}^m (f(x_i) - g(x_i))^2} \right] = 0 \quad (3.2)$$

where,

- x_i = data points
- $f(x_i)$ = certified arrived population density function
- $g(x_i)$ = conceptual SDI Architecture population arrival population density function
- h = sampling interval

From the research area of neural networking, specifically dealing with the field theory of self organizing neural nets, by Amari [3], detectors, like the UII comparators, enabling the type of network self organization in the brain, enabled by the adaptive routing of the UII Algorithm preserve the topological structure operation of the network.

REFERENCES

- [1] A.K. Aho, J. E. Hopcroft, J. D. Ullman, Data Structures and Algorithms, Addison Wesley, June 1983.
- [2] A. Albanese, "Star Network with Collision-Avoidance Circuits", IEEE Global Telecommunication Conference, San Diego, California, Nov. 28-Dec. 1, 1983.
- [3] S. Amari, "Field Theory of Self-Organizing Neural Nets", IEEE Trans On Systems, Man and Cybernetics, Vol. SMC-13, No. 5, September/October, 1983, pp. 741-747.
- [4] K. M. Baumgartner, and B. W. Wah, "The Effects of Load Balancing on Response Time for Local Computer Systems with a Multiaccess Network", IEEE International Conference on Communications, pp. 262-266, June 23-26, 1985.
- [5] R. Bayer, H. Heller, A. Reiser, "Parallelism And Recovery in Database Systems", ACM Transactions on Database Systems, Vol. 5, No. 2, June 1980, pp. 139-156.
- [6] D. P. Bertsekas, "Dynamic Models of Shortest Path Routing Algorithms For Communication Networks with a Ring Topology", Coordinated Science Laboratory Working Paper, University of Illinois at Urbana, Champaign, September, 1978.
- [7] L. D. Cheremisinova, "Implementation of Parallel Digital Control Algorithm by Asynchronous Automata", Automatic Control and Computer Science, Vol. 19, No. 2, pp. 65-69, 1985.
- [8] R. C. Dixon, "Lore of the Token Ring", IEEE Network Magazine, Vol. 1, No. 1, pp. 11-18 January, 1987.
- [9] A. Geissler, J. Hanle, A. Konig, E. Pade, "Free Buffer Allocation - An Investigation by Simulation", Computer Networks, North Holland Publishing Co., Vol. 2, pp. 191-208, 1978.
- [10] I. I. Gikhman, A. V. Skorokhod, "Introduction to Theory of Random Process", p. 319.
- [11] G. H. Gonnet, P. A. Larson, "External Hashing with Limited Internal Storage", Technical Report CS-82-38, University of Waterloo, 1982. Extended abstract in Proc. ACM Symposium on Principles of Database Systems, ACM, New York, 1982.

REFERENCES (Continued)

- [12] D. A. Huffman, "The Design and Use of Hazard-Free Switching Networks," JACM, Vol. 4, No. 41, January 1957, pp. 47-62.
- [13] S. F. Ivochkina and V. G. Litvin, "Approximate Analysis of Hierarchical Information and Computing Systems and Blocking", Automatic Control and Computer Science, Vol. 19, No. 2, pp. 37-42, 1985.
- [14] Yu.G.Karpov and A. V. Tolstyakov, "Analytic Method of Protocol Verification", Avtomatika i Vychislitel'naya Tekhnika, Vol. 19, No. 1, pp. 35-41, 1985.
- [15] I. V. Kogan, I. I. Kogan, "Hierarchical Representation Of Algorithms For Computer Network Models", Avtomatika i Vychislitel'naya Tekhnika, Vol. 19, No. 1, pp. 29-34, 1985.
- [16] V.Ya.Kontorovich, "Problems of Investigation of Dynamic Systems with A Random Structure", Avtomatika i Vychislitel'naya Tekhnika, Vol. 19, No. 2, pp. 21-26, 1985.
- [17] Per-Ake Larson, "Hash Files: Some Recent Developments", Proceedings of the First International Supercomputer Conference, February, 1985, pp. 671-679.
- [18] D. Lazak, "User Oriented Optimal Packet Sizes within Packet Switching Systems", Angewandte Informatik 6/79, pp. 259-265.
- [19] A. A. Lazard, "Optimal Flow Control of An M/M/m Quene", "Journal of the Association for Computing Machinery, Vol. 31, No. 1, January, 1984, pp. 86-98.
- [20] V. I. Levin, "Analysis of Loading of Computer System" Automatic Control and Computer Science, Vol. 17, No. 6, pp. 67-73, 1983.
- [21] V. I. Levin, "Logical Methods of Investigating Real Time Computer System", Automatic Control and Computer Science, Vol. 19, No. 3, pp. 24-30, 1985.
- [22] V. G. Litvin, "Diffusion Model of a Multiprocessor Computer System", Automatic Control and Computer Science, No. 4, pp. 48-52, 1983.
- [23] G. A. Maley, J. Earl, Logic Design of Digital Circuits. W. L. Everitt, Editor, Prentice Hall EE Series, International Series in Electrical Engineering, pp. 198, 213, 216, 217, 250-254.

REFERENCES (Continued)

- [24] N. M. Nikitin, S. L. Okunev and E. A. Saksonov, "Contention Resolution Algorithm for a Local Area Network with Random Multiple Access", Automatic Control and Computer Science, Vol. 19, No. 5, pp. 41-46, 1985.
- [25] A. A. Nilsson and W. Chou, "General Performance Model for Different Ring Network Protocols", pp. 1-13.
- [26] C. A. Niznik, "A Congestion-Capacity Product for Computer Communication Networks", Journal of Combinatorics, Information and System Sciences, Vol. 4, No. 3, pp. 205-210, 1979.
- [27] C. A. Niznik, "Performance Evaluation Of The Computer Network Dynamic Congestion Table Algorithm", IEEE Transactions On Computers, Vol. C-33, No. 2, February, 1984.
- [28] C. A. Niznik, "The Gateway Congestion Control Policy, A Congestion Monitored Software for Computer Internetworks," Proceedings of the First International Supercomputer Conference, February, 1985.
- [29] C. A. Niznik, C. Watts, W. Hoss, "Mathematical Entropy Formalism For Robotic Neural System Quantization Theory", Proceedings of IEEE International Symposium On Intelligent Control, January, 1987.
- [30] G. J. Nutt, "Evaluation Nets For Computer System Performance Analysis", Fall Joint Computer Conference, pp. 279-286, 1972.
- [31] A. Yavuz Oruc, "A Classification of Cube-Connected Networks with a Simple Control Scheme", IEEE Transactions on Computers, Vol. C-33, No. 8, August 1984, pp 769-772.
- [32] C. H. Panadimitriov, "Theory of Concurrency Control", Lecture notes in Computer Science, Theoretical Computer Science, No. 145.
- [33] J. L. Peterson, Operating System Concepts, second edition, Addison Wesley, 1985.
- [34] I. A. Pogosyan, "One Problem of Calculation of Efficiency Figures for Shared-Use Computer Centers", Automatic Control and Computer Science, Vol. 14, No. 3, pp. 52-57, 1980.
- [35] A. Ralston (editor), C. L. Meek (Asst.editor), Encyclopedia of Computer Science, first edition, Petrocelli/Charter, A., New York 1976, pp. 604-607.

REFERENCES (Continued)

- [36] D. J. Rosenkrantz, R. E. Stearns and P. M. Lewis II, "System Level Concurrency Control For Distributed Database System", ACM Transaction on Database Systems, Vol. 3, No. 2, pp. 178-198, June 1978.
- [37] F. A. Sklyarevich, "Residence Characteristics of a Network Fragment in a Fixed Subset of States", Automatic Control and Computer Science, Vol. 19, No. 2, pp. 14-20, 1985.
- [38] N. C. Strole, "The IBM Token-Ring Network - A Functional Overview", IEEE Network Magazine, Vol. 1, No. 1, pp. 23-30, January, 1987.
- [39] S. P. Sushchenko, "Effect of Buffer Blocking on the Operating Characteristics of a Data Link", Avtomatika i Vychislitel'naya Tekhnika, Vol. 19, No. 6, pp. 27-34, 1985.
- [40] S. P. Sushchenko, "Method of Choosing Efficient Packet Length in Packet-Switched Networks", Avtomatika i Vychislitel'naya Tekhnika, Vol. 18, No. 3, pp. 24-28, 1984.
- [41] S. S. Thakkar, A. E. Knowles, "Virtual Address Translation Using Parallel Hashing Hardware", Proceeding of the First International Supercomputer Conference, February, 1985, pp. 697-705.
- [42] S. Toueg, "The Design of Deadlock and Livelock Free Packet Switching Networks", Ph.D. Dissertation, Dept. of Electrical Engineering, Princeton University, 1979.
- [43] P. P. Treis, "Collision Free Star Broadcast Medium", Automatic Control and Computer Science, Vol. 19, No. 6, pp. 42-47, 1985.
- [44] P. F. Walker, "Smart Weapons In Naval Warfare", Scientific American, Vol. 248, No. 5, May 1983, pp. 53-61.
- [45] M. Willet, "Token Ring Local Area Networks - An Introduction", IEEE Network Magazine, Vol. 1, No. 1, pp. 8-10, January, 1987.
- [46] G. F. Yanbykh, "Optimum Placement of Remote Information Concentrators in a Centralized Data Transmission Network Taking Into Account The Reliability Of Communication Channels", Avtomatika i Vychislitel'naya Tekhnika, Vol. 18, No. 3, pp. 44-47, 1984.
- [47] Takshing P. Yum and Hua-Chun Liu, "Adaptive Load Balancing For Parallel Queues", Links for the Future, Science, Systems and Services for Communications, P. Dewilde and C. A. May (editors), IEEE/Elsevier Science Publisher B. V. (North Holland), pp. 1268-1271, 1984.

REFERENCES (Continued)

- [48] V. N. Zablotskii, V. E. Spasskii, E. F. Venzel and N. J. Goroburdo, "Approximate Analytical Model of Multiprocessor Computer System with a Ring Structure", Automatic Control and Computer Science, Vol. 18, No. 3, pp. 44-47, 1984.
- [49] E. V. Zinov'ev, A. A. Strekalev, "Control of Distributed Transactions in a Computer Network", Avtomatika i Vychislitel'naya Tekhnika, Vol. 20, No. 1, pp. 61-72, 1986.

ACRONYMS AND ABBREVIATION

AATPE	Advance Acquisition, Tracking, and Pointing Experiment
ABL	Airborne Laser
ABM	Antiballistic Missile
ACM	Attitude Control Module
ACP	Airborne Command Post
ACS	Attitude Control System
Ada PDL	Ada Process Description Language
ADC	ANALOG to Digital Converter
ADI	Air Defense Initiative
ADIO	ADCOM Intelligence Officer
ADOC	Air Defense Operations Center
ADOP	Advanced Distributed On-Board Processor
ADPE	Automated Data Processing Equipment
AEM	Arsenal Exchange Module
AERIE	Aerial Intercept Element
AFB	Air Force Base
AFS	Air Force Station
AFSC	Air Force Systems Command
AFSPACECOM	Air Force Space Command
AFSTC	Air Force Science Technology Center
AI	Artificial Intelligence
AIAA	American Institute of Aeronautics and Astronautics
AJ	Antijam
ALCC	Airborne Launch Control Center
ALCM	Air-Launched Cruise Missile
ALERTCON	Alert Condition
ALS	Advanced Launch System
AMOS	Air Force Maui Optical Tracking Station
ANMCC	Alternate National Military Command Center
AOA	Airborne Optical Adjunct
AOS	Airborne Optical System
AOSP	Advanced On-Board Signal Processor
APA	Antenna Point and Acquisition

APU	Auxiliary Power Unit
ARTS	Automated Remote Tracking Station
ASAT	Antisatellite
ASDP	Advanced Sensor Demonstration Program
ASM	Air to Surface Missile
ASW	Antisubmarine Warfare
ASWCCS	Antisubmarine Warfare Command and Control System
ASWOC	Antisubmarine Warfare Operations Center
ATCC	Air Traffic Control Center
A/VLSI	Advanced Very Large Scale Intergration
AW/AA	Attack Warning/Attack Assessment
AWACC	Airborne Warning and Control Center
AWACS	Airborne Warning and Control System
AWDS	Automated Weather Distribution System
AWPDS	Attack Warning Processing and Display System
BBL	Brassboard Level
BCD	Baseline Concept Definition
BeH ₂	Beryllium Hydride
BER	Bit Error Rate
BIC	Braduskill Interceptor Concept
BIS	Bank of International Settlements
BM	Battle Manager
BM/C ³	Battle Management/Command, Control, and Communications
BMD	Ballistic Missile Defense
BMDOC	Ballistic Missile Defense Operations Center
BMEWS	Ballistic Missile Early Warning System
BP	Boost Phase
BPA	Budget Plan Authority
B/PBTBM	Boost/Post Boost Tier Battle Manager
BQ	Beam Quality
BRDF	Bidirectional Reflectance Distribution Function
BSTS	Boost Surveillance and Tracking System
C ²	Command and Control
C ³	Command, Control, and Communications

C ³ I	Command, Control, Communications and Intelligence
CC	Carbon-Carbon
CDR	Critical Design Review
CDRL	Contract Data Requirements List
CELV	Complementary Expendable Launch Vehicle
CEP	Circular Error Probability
CER	Cost Estimating Relationship
CETD	Concept Experiment Test and Development
CHOP	Change in Operational Procedures
CINC	Commander-in-Chief
CINCSD	Commander-in-Chief Strategic Defense
CINCSpace	Commander-in-Chief U.S. Space Command
CJCS	Commander Joint Chiefs of Staff
CIF5	Chlorine Pentafluoride
CM	Countermeasure
CMBA	Cheyenne Mountain Complex
CMEA	Council for Mutual Economics Assistance
CMOS	Complementary Metal Oxide Semiconductor
CO ₂	Carbon Dioxide
COBRA DANE	Shemya, AK Tracking Radar System
COMSAT	Communication Satellite
COMSEC	Communication Security
CONSU	Continental Soviet Union
CORA	Coherent Optical Radar Amplifier Program
CP	Command Post
CPC	Coherent Power Combiner
CPI	Consumer Price Index
CPSU	Communist Party of the Soviet Union
CSOC	Consolidated Space Operations Center
CSO	Closely Spaced Objects
CV	Carrier Vehicle
CW	Continuous Wave
DAA	Direct Ascent ASAT
DAB	Defense Acquisition Board
DABM	Defense Against Ballistic Missiles

DANA	Direct Ascent Nuclear ASAT
DANNKA	Direct Ascent Nonuclear Kill ASAT
DARPA	Defense Advanced Research Project Agency
DASAT	Direct-Ascent Antisatellites
DD&T	Design, Development, and Test
DE	Damage Expectancy
DEFCON	Defense Condition (JCS)
Dem/Val	Demonstration/Validation
DEW	Directed Energy Weapon
DEW	Distant Early Warning
DF	Deuterium Fluoride
DIA	Defense Intelligence Agency
DIDS	Dynamic Interdiction Discrimination System
DMP	Defense Machinery Production
DMS	Data Management System
DMSP	Defense Meteorological Satellite Program (METSAT)
DOD	Department of Defense
DSAT	Defense Satellite
DSCS	Defense Satellite Communications System
DSP	Defense Support Program
DST	Defense Suppression Threat
DT&E	Development, Test and Evaluation
DT&V	Development, Test, and Validation
DYSC	Dynamic Spherical Coordinate
EAM	Emergency Action Message
ECC	Emergency Command Cell
ECCM	Electronic Counter-Countermeasures
ECL	Emitter Coupled Logic
ECM	Electronic Countermeasure
ECMC	Enhanced Crisis Management Capability
ECS	Environmental Control System
EDAC	Error Detection and Correction
EKF	Extended Kalman Filter
EM	Electromagnetic
EMC	Electromagnetic Compatibility

EMI	Electromagnetic Interface
EML	Electromagnetic Launcher
EMP	Electromagnetic Pulse
EMRLD	Exier orate Power Raman-Shifted Laser Devise
EMSSP	Electrogetic Standards and Specification Program
ENDO	Endoatmosphere
EOCM	Electro-Optic Countermeasure
EPOS	Engineering and Project Management Oriented Support System
EPS	Electric Power System
ERCS	Emergency Rocket Communication System
ERINT	Emergency Rocket Interceptor
ERIS	Exoatmospheric Reentry Interceptor Subsystem or System
ETEEM	End-To-End Engineering Model
EW	Electronic Warfare
EWR	Early Warning Radar
FBB	Fast Burn Booster
F/C	Fire Control
FCDL	Fire Control Data Link
FDA	Final Documentation and Analysis
FEL	Free Electron Laser
FLAGE	Flexible Lightweight Agile Guided Experiment
FO	Follow-On
FOB	Fractional Orbital Bombardment System
FOC	Full Operational Capability
FOG	Fiber Optics Gyro
FOR	Field of Regard
FORCECOM	U.S. Force Command
FOT&E	Follow-On Operational Tests and Evaluation
FOV	Field of View
FPA	Focal Plane Array
FSD	Full Scale Development
FSED	Full Scale Engineering Development
FT	Fault Tolerance
FTD	Foreign Technology Division
FY	Fiscal Year

FYP	Five Year Plan (Soviet)
G&C	Guidance & Control
G/AIRT	Ground/Air Interface Terminal
GATT	General Agreement on Trade and Tariff
GBL	Ground-Based Laser
GBLRS	Ground-Based Laser Repeater Station
GBLU	Ground-Based Laser Uplink
GBOS	Ground-Based Optical System
GBPS	Giga Bits Per Second
GBPST	Ground-Based Passive Signal Tracking
GBR	Ground-Based Radar
GDR	German Democratic Republic
GEO	Geosynchronous Orbit
GEODSS	Ground-Based Electro-Optical Deep Space Surveillance System
GHz	Gigahertz
GLCM	Ground-Launched Cruise Missile
GLOW	Gross Liftoff Weight
GMCP	Ground Mobile Command Post
GNP	Gross National Product
GOPS	Giga (billion) Operations Per Second
GPPS	Giga Pluses per Second
GPS	Global Positioning System
GSE	Ground Support Equipment
GSTS	Ground Surveillance and Tracking System
GVSC	Generic VHSIC Spaceborne Computer
GW	Gigawatts
GaAs	Gallium Arsenide
HA	Higher Authority
H/W	Hardware
HEDI	High Endoatmospheric Defense Interceptor
HEDS	High Endoatmospheric Defense System
HEMT	High Electron Mobility Transistor
HF	Hydrogen Fluoride
HIBREL	High Brightness Relay

HICAMP	Highly Calibrated Airborne Measurement Program
HIP Be	Hot-Isostatic Press Beryllium
HLG	Hemispherical Laser Gyro
HLLV	Heavy Lift Launch Vehicle
HLV	Heavy Lift Vehicle
HOL	High Order Language
HSSU	High-Speed Search Unit
HTK	Hard Target Kill
HYVINT	Hyper-Velocity Interceptor
I&W	Indications and Warnings
IAW	In Accordance With
ICBM	Intercontinental Ballistic Missile
ICT	Intelligence Cycle Time
ID	Identification
IDHS	Intelligence Data Handling System
ILDC	Integrated Logic Design Concepts
IMF	International Monetary Fund
IMU	Inertial Measurement Unit
INF	Intermediate Nuclear Forces
IOC	Initial Operating Capability
IPP	Impact Point Prediction
IR	Infrared
IR&D	Internal Research and Development
IRBM	Intermediate Range Ballistic Missile
ISP	Specific Impulse
ITUE	Integrated Test Uplinks Experiment
ITW&A	Integrated Tactical Warning and Assessment
IUS	Inertial Upper State
JCS	Joint Chiefs of Staff [also nuclear hardness level specification (i.e., JCS XXX)]
JSS	Joint Surveillance System
JSTPS	Joint Strategic Target Planning Staff
KA	Kill Assessment
KBPS	Kilo Bits Per Second

KED	Kill Enhancement Device
KEW	Kinetic Energy Weapon
kg	Kilogram
KKV	Kinetic Kill Vehicle
KREMS	Kiernam Re-Entry Measurement Site
KSC	Kennedy Space Center
KT	Kilo Ton
KV	Kill Vehicle
KW	Kilo Watts
LADAR	Laser Radar
LANTFLEET	Atlantic Fleet
LAU	Launch
LCC	Life Cycle Cost
LDC	Less-Developed Country
LDR	Leadership
LEO	Low Earth Orbit
LICD	Laser Imaging Component Development
LMF	Large Mirror Facility
LPARDS	Large Phase Array Radar Detection System
LPP	Launch Point Prediction
LRB	Liquid Rocket Booster
LRU	Line Replacement Unit
LUA	Launch-Under-Attack
LWIR	Long Wavelength Infrared
M ₁	Basic Money Supply
MAC	Military Airlift Command
MANTECH	Manufacturing Technology
MaRV	Maneuvering Reentry Vehicle
MBMW	Machine Building and Metal Working
MBPS	Mega Bits per Second
MC	Midcourse Phase
MCC	Mission Control Center (Satellite)
MCP	Mobile Command Post
MDP	Mission Data Processing

METSAT	Meteorological Satellite
MFC	Matched Filter Correlator
MILSTAR	Military Extremely High Frequency (EHF) Satellite Communications System
MIPS	Million Instructions Per Second
MIRV	Multiple Independently Targeted Reentry Vehicles
MIS	Missile
MLC	Mobile Liaison Cell
MMH	Monomethylhydrazine
MMW	Millimeter Wave
MOA	Memorandum Of Agreement
MOE	Measure of Effectiveness
MOM	Measure of Merit
MOPA	Master Oscillator Power Amplifier
MOPS	Million Operations Per Second
MOTR	Multiple Object Tracking Radar
MOU	Memorandum Of Understanding
MRP	Multiple Reentry Payload
MRPKKV	MRP Kinetic Kill Vehicle
MS	Milestone
MISC	Missile and Space Intelligence Center
MSLOC	Millions of Source Lines Of Code
MSSTM	Military Space System Technology Model
MUX	Multiplex
MW	Megawatts
MWC	Missile Warning Center
MWDS	Missile Warning and Display System
MWIR	Mid-Wavelength Infrared
N ₂ H ₄	Hydrazine
N ₂ O ₄	Nitrogen Tetroxide
NASA	National Aeronautics and Space Administration
NASP	National Aerospace Plane
NATO	North Atlantic Treaty Organization
NAVSPASUR	Naval Space Surveillance System
NCA	National Command Authority
NCCS	Navy Command and Control System

NDEW	Nuclear Directed Energy Weapon
NDS	Nuclear Detection System (formerly IONDS)
NEACP	National Emergency Airborne Command Post
NEP	National Economic Plan
NIO	National Intelligence Officer
NKEW	Nuclear Kinetic Energy Weapon
NM	Network Manager
NMCC	National Military Command Center
NMCS	National Military Command System
NMP	Net Material Product
NOP	Nuclear Operations Plan (NATO)
NORAD	North American Aerospace Defense Command
NPB	Neutral Particle Beam
NPES	Nuclear Planning and Execution System
NTF/NTB	National Test Facility/National Test Bed

O&M	Operations & Maintenance
O&S	Operations and Support
OA	Oscillator Amplifier
OAI	Oscillator Amplifier Integration
OAMP	Optical Airborne Measurement Program
OB	Order of Battle
OBP/NM	On-Board Processor for Network Manager
OFVR	Out-of-Field-of-View Rejection
OIS	Orbit Insertion System
OMT	Other Military Targets
OMV	Orbital Maneuvering Vehicle
OSD	Office of the Secretary of Defense
OSIS	Ocean Surveillance Information System
OT&E	Operational Test and Evaluation
OTH	Over-The-Horizon
OTH-B	Over-the-Horizon Backscatter Radar

PACCS	Post-Attack Command Control System
PARCS	Perimeter Acquisition Radar System
PATHS	Precursor Above-The-Horizon Sensor

PAVE PAWS	SLBM Early Warning System
PAXBAR	Pacific Radar Barrier
PB	Post-Boost Phase
PBALL	Precision Ballistics
PBV	Postboost Vehicle
PDR	Preliminary Design Review
PINS	Point In Space
PK	Probability of Kill
POS	Positioner Stage
PtSI	Platinum Silicon
PV-HgCdTe	Photovoltaic Mercury Cadmium Telluride
R ³	Rotary Reciprocating Refrigerator
R&D	Research and Development
RAD	Radiation Absorbed Dose
RAM	Radar Absorption Material
RAM	Random Access Memory
RDT&E	Research Development Test and Evaluation
RECCE	Reconnaissance
REM	Radiation Equivalent Man
RF	Radio Frequency
RFW	Radar Frequency Weapon
RID	Range Insensitive Direction
RLG	Ring Laser Gyro
RMA	Reliability / Maintainability/Availability
ROCC	Regional Operations Control Center
ROE	Rule Of Engagement
ROFT	Rapid Optic Fabrication Technology
ROM	Read-Only Memory
RU	Rubles
RV	Reentry Vehicle
RVF	RV Finder
RV FINDER	Reentry Vehicle Flash Identification and Discrimination Evaluation Research
S & SED	Satellite & Space Electronics Division, Rockwell International
SAC	Strategic Air Command

SACC	Sector Antisubmarine Control Center
SACDIN	Strategic Air Command Digital Information Network
SADMT	SDI Architecture Dataflow Modeling Technique
SALT	Strategic Arms Limitation Treaty
SAM	Surface-to-Air Missile
SAMTO	Space and Missile Test Organization
SATKA	Surveillance Acquisition, Tracking, and Kill Assessment
SBFEL	Space-Based Free-Electron Laser
SBI	Space-Based Interceptor
SBL	Space-Based Laser
SBSP	Space-Based Support Platform
SCF	Satellite Control Facility
SCIS	Survivable Communications Integration System
SCN	Space Communications Network
SCP	System Concept Paper
SDI	Strategic Defense Initiative
SDIO	Strategic Defense Initiative Organization
SDLV	Shuttle-Derived Launch Vehicle
SDR	System Design Review
SDS	Strategic Defense System
SDS-CC	Strategic Defense System Command Center
SDS-OC	Strategic Defense System Operations Center
SECC	Survivable and Enduring Command Center (SAC)
SEER	Sensor Experimental Evaluation and Review
SEO	Survivability Enhancement Option
SEWS	Satellite Early Warning System
SHAPE	Supreme Headquarters Allied Power Europe (NATO)
SHF	Super High Frequency
SIE	SATKA Integrated Experiment
SIOP	Single Integrated Operational Plan
SIOP CINC	U&S Commander with Forces Committed to the SIOP
SIT	Spacecraft Integration and Test
SITREP	Situation Report
SLBM	Submarine Launched Ballistic Missile
SLC	Space Launch Complex
SLCM	Sea-Launched Cruise Missile

SLOC	Source Lines of Code
SMATH	Satellite Material Hardening (standardized)
SNF	Strategic Nuclear Forces
SOA	State Of Art
SOC	Satellite Operations Center/Sector Operations Center
SOCCC	Satellite Operations Center/Submarine Operational Control Center
SOF	Strategic Offensive Forces
SOSUS	Sound urveillance System
SOVA	Soviet Analysis
SPACETRACK	AF Satellite Surveillance Network
SPADCCS	Space Defense Command and Control Center
SPADOC	Space Defense Operations Center
SPETSNAZ	Covert Soviet Special Forces
SPOC	Space Operations Center
SPR	Spares
SQM	Space-Qualified Model
SRAM	Short-Range Attack Missile
SRB	Solid Rocket Booster
SSBN	Ballistic Missile Nuclear-Energy Submarine
SSC	Space Surveillance Center
SSGN	Cruise Missile Nuclear-Energy Submarine
SSME	Space Shuttle Main Engine
SSN	Space Surveillance Network
SSP	Small Sortie Payload
SSP	Solid State Photomultiplier
SSTS	Space Surveillance & Tracking System
STAR	System Threat Assessment Report
STAS	Space Transportation Architecture Study
STP	Space Test Program
STR	Strategic Forces
STS	Space Transportation System
SU	Soviet Union
SURTASS	Surface Tower Array Surveillance System
SV	State Vector
SVEC	Surveillance, Vulnerability, Endurability and Connectivity
S/W	Software

SWHCC	Static War Headquarters Command Center
SWIR	Short Wavelength Infrared
TAC	Tactical Air Command
TAV	Transatmospheric Vehicle
TBD	To Be Determined
TBM	Tactical Ballistic Missile
TBR	To Be Refined
TCN	Terrestrial-Based Communications Network
TDI	Time-Delay Integration
TDRSS	Tracking and Data Relay Satellite System
TEL	Transported Erector Launcher
TFU	Theoretical First Unit Costs
TIR	Terminal Imaging Radar
TMD	Theater Missile Defense
TOF	Time of Flight
TP	Terminal Phase
TPE	Tracking and Pointing Experiment
TRANSCOM	Transportation Command
TRANSIT	Navy Navigation Satellite
TT&C	Telemetry, Tracking, and Communications
TTI	Time-to-Impact
TW/AA	Threat Warning/Attack Assessment
TWT	Travelling Wave Tube
U&S Command	Unified and Specified Commands
UCP	Unified Command Plan
UNAAF	Unified Action Armed Forces (JSC Pub 2)
USCINCEUR	U.S. Commander-in-Chief Europe
USCINCLANT	U.S. Commander-in-Chief Atlantic
USCINCPAC	U.S. Commander-in-Chief Pacific
USEURCOM	U.S. European Command
USLANTCOM	U.S. Atlantic Command
USPACOM	U.S. Pacific Command
USSOUTHCOM	U.S. South Command
USSPACECOM	U.S. Space Command

USSR	Union of Soviet Socialist Republic
U.S.	United States
UV	Ultraviolet
VAFB	Vandenberg Air Force Base
VHSIC	Very High-Speed Integrated Circuit
WBM	Weapon Battle Manager
WIS	WWMCCS Information System
WPI	Wholesale Price Index
WSI	War Support Industry
WWMCCS	Worldwide Military Command and Control System
XRL	X-Ray Laser