Dynamically Reconfigurable Networks: Concept Evaluation Through Simulation

Anjali Venkateshwaran, Richard E. Nance and Osman Balci

TR 88-34

Technical Report SRC 88-003

DYNAMICALLY RECONFIGURABLE NETWORKS:

CONCEPT EVALUATION THROUGH SIMULATION

Anjali Venkateshwaran
Richard E. Nance
Osman Balci

Systems Research Center
and
Department of Computer Science
Virginia Polytechnic Institute & State University
Blacksburg, Virginia 24061
USA

January 1988

This research was supported in part by the Naval Sea Command under Contract No. N60921-83-G-A165-B032 through the Systems Research Center at Virginia Tech

Abstract

The dynamically reconfigurable hierachical network offers decided advantages in reliability and survivability. The concept is evaluated strictly in terms of message transmission delays for demand scenarios that vary widely. A SIMULA 67 model provides the experimental basis for obtaining preliminary results that show the reductions in transmission delays to be significant and to increase with the size of the network.

CR Categories and Subject Descriptors: C.2.1. [**Computer-Communication Networks**]: Network Architecture and Design -- *distributed networks, network topology*; C.2.5 [**Computer-Communication Networks**]: Local Networks; I.6.4 [**Simulation and Modeling**]: Model Validation and Analysis.

General Terms: Performance, Verification.

Additional Key Words and Phrases: Message Transmission Delay, Dynamic Reconfigurability.

Introduction

The concept of dynamic reconfigurability is not new. Self-adaptive operating systems were research topics in the 1970s [1], well before "intelligent systems" became the dominant buzzword. The motivation for dynamic reconfigurability in local area networks (LANs), advanced for packet switching hierarchical networks by Nance in 1979 [2], stems from the need to serve an application which exhibits extremes in message traffic demands. Characterized by an underlying command hierarchy, in addition to high variability in message traffic, the naval surface combatant requirements spawned the *dynamic hierarchy concept* as an approach to improve survivability, adaptability, and reliability. Clearly, the *fully distributed system* [3] was inappropriate for the naval application since the acquisition of global knowledge was a mandatory requirement.

A dynamic hierarchy is a LAN architecture that utilizes a strictly hierarchical topology (a tree) but the apex (root of the tree) can be assigned to any of a set of candidate nodes, and both connectivity and channel bandwidth can be reallocated. The architecture is described in a tutorial fashion in [4]; a model of link capacity assignment is given in [5]; and a procedure for evaluating heuristic techniques for bandwidth assignment decisions is provided in [6]. More recent studies of the concept have sought to develop analytic (matrix-geometric) [7,8] and simulation [9,10] models to evaluate the requisite cost and benefits of dynamic reconfigurability for such networks.

The objectives of this paper are to explain the architectural characteristics of dynamic hierarchical local area networks (DHLANs), describe a simulation model to assess the cost and benefits of dynamic reconfigurability, and present some preliminary, but promising, results. Section 2 describes the DHLAN architecture and the protocol for dynamic reconfiguration. Section 3 treats the simulation model development and the verification procedures applied to the model. Preliminary results are presented in the fourth section, followed by a brief summary and conclusions.

Dynamic Hierarchical Networks

Dynamic reconfigurability can take on progressively more inclusive forms. The limited form examined to date and the particulars of a protocol to implement reconfiguration requirements are described below.

The DHLAN Architectural Concept

The dynamic hierarchy is a generalization of the conventional tree-structured network, which mandates a centralized, strictly hierarchical mode of control over link message traffic among nodes. The generalizations are embodied in the potential reconfiguration of the network, in response to external or internal conditions, that permits:

(1) designation of a new apex from among the predefined Apex Candidate Set (ACS),

(2) redefinition of connectivity through the activation of inactive links,

(3) reassignment of capacity among active links to effect better service for anticipated changes in message demands.

These three reconfiguration properties represent increasing degrees of modification and requisite technical difficulty; however, all three can be achieved within current technology.

The ability for apex redesignation -- a new base of control-- contributes significantly to improved reliability in response to internal node failures and to increased survivability and adaptability through added potential for degraded mode operation. The common uses of redundant links, "hot spares", and redefined routings can be employed to provide connectivity alternatives, which lead to the important distinction between *logical* and *physical* topologies. The same possibilities and others provided by satellite communications and fiber optics technology enable the reassignment of bandwidth (link capacity).

The Reconfiguration Protocol

The network service is conveniently divided into three temporal periods: (1) *regular operation*, all messages transmitted according to prescribed routings, (2) *reconfiguration*, messages among ACS nodes are buffered while "local hierarchies" continue as usual, and (3) *adjustment*, the backlog of buffered messages is gradually eliminated as current demands are met. Figure 1 depicts the topological changes during each period.

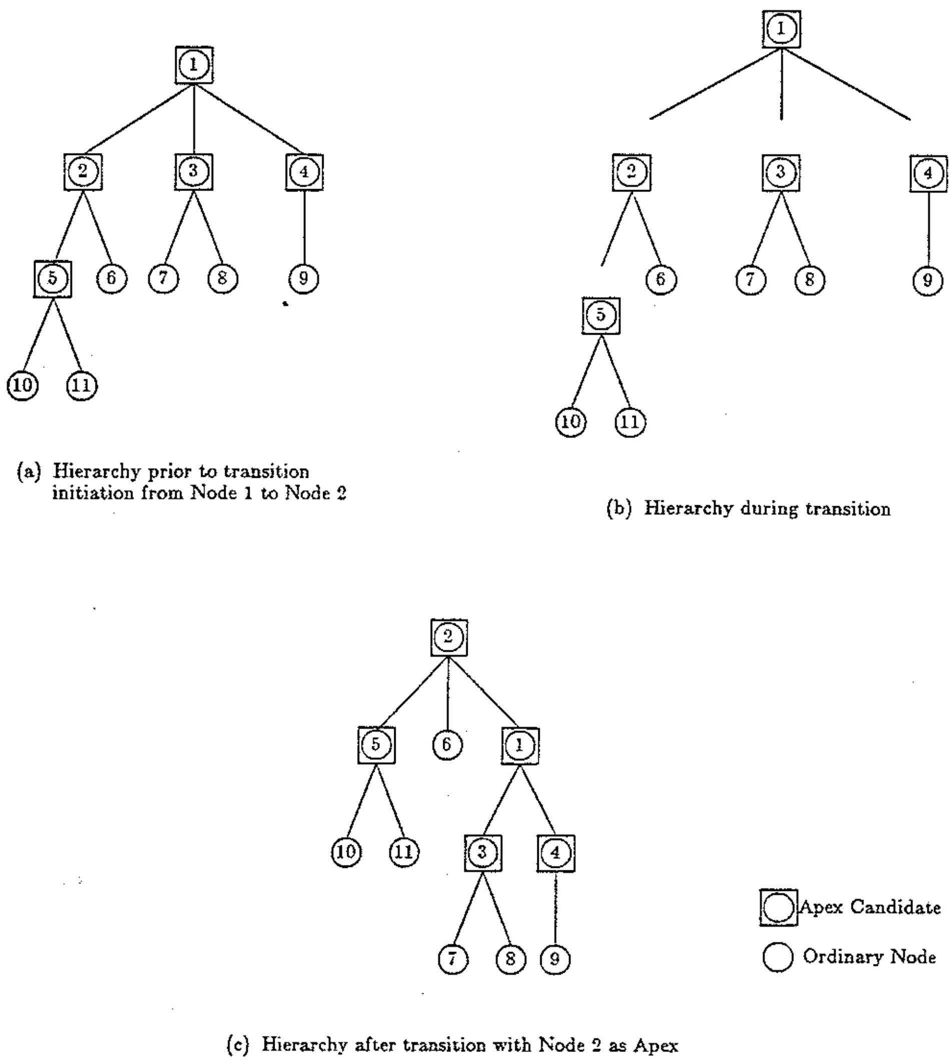


Figure 1. Network Topological Structure During Transition (from [9, p.22])

The need for reconfiguration is assumed to be conveyed by message -- an event signal. Apex transition is accomplished using the following protocol:

(1) The first ACS node to recognize the event (the initiating node) sends a *transition request* to the incumbent apex.

(2) The incumbent apex sends a request to effect a transition (a *transition vote*) to all ACS members, which alerts the new apex of its intended role.

(3) A *transition acknowledge* is sent by each ACS node to inform the incumbent that a transition is accepted. No acknowledgment represents an unaccepted decision.

(4) The incumbent apex, based on ACS responses, can abort the transition by sending a *transition over* to all ACS nodes or effect the transition by sending a *transition commit* to all ACS nodes.

(5) Each ACS node, on receipt of the *transition commit,* interrupts message transmission to all ACS nodes and buffers these messages until the reconfiguration period ends. Each ACS node with a "local hierarchy" continues the local service.

(6) Active routing tables are updated and link capacity assignment decisions are executed. A *transition over* is sent by both the former and the new apex, with the receipt of both signaling the termination of the reconfiguration period for that ACS node.

Defining the protocol to this level has been necessary to estimate the delay incurred in reconfiguration for modeling purposes. More detailed experimentation with protocol definition is expected with an experimental network configuration.

A Simulation Model for Concept Evaluation

Venkateshwaran [10] cites 15 simulation studies of LANs, calling attention to the wide variations in objectives and implementation languages. While two studies have comparable objectives [11, 12], no effort has addressed the evaluation of the dynamic reconfigurability concept. Prior studies of LAN applications within naval systems, such as [13], have restricted their focus to particular sets of parameters, with values remaining unchanged throughout a run.

Overview of the Simulation Model

Figure 2, taken from [10, p. 28] conveys the structure of the DHLAN model. Following [14], the network model is comprised of a set of nodes, a set of links, a configuration change initiator, and two output processes for data collection and analysis. A node is an abstract data type that includes a *receive process*, two *transmit processes*, and two queues associated with each process, one for control (CQ) and the other for data (MQ) messages. The distinctions among current apex, remaining ACS and "ordinary" nodes is found in the specification of the transmit and receive processes. Each node has a defined routing table, and a message generator to produce local demand is also defined for each node. For the purposes of the study reported here, no distinction is made for message types and the physical topology (and connectivity) remain invariant during a run.

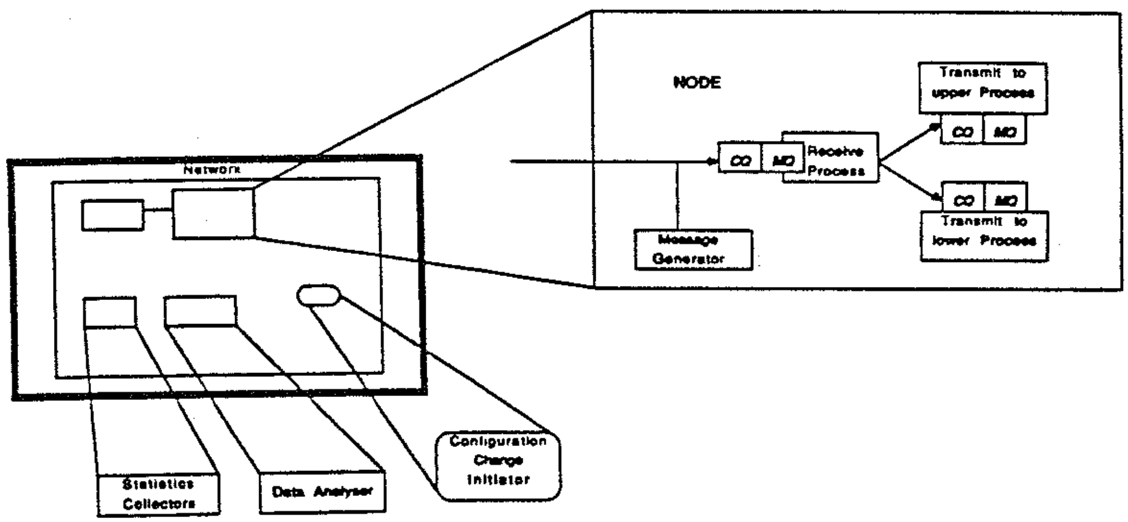


Figure 2. Structure of the DHLAN Simulation Model [9, p.31].

The configuration changer initiates the messages leading to reconfiguration following a deterministic or random specification of inter-event times. The statistics collector accumulates delay components for each message and provides information on queue contents. The initialization procedure confines all the data structures and operations for initial state definition within a single process.

Examining the requirements associated with concept evaluation, i.e. the correctness, adaptability, and testability of the simulation model, the project team felt SIMULA to be the preferred choice. Team members have fluency in other simulation programming languages, notably GPSS/H, SLAM II, and Simscript II.5. In fact, during an interruption in availability of the SIMULA compiler, another project member produced an equally complex Simscript model (1075 lines of source code and documentation) for the cross-validation of an analytic model [8].

Model Verification Procedures

Perhaps the key element in the verification of the DHLAN model is the assignment of verification responsibilities to a team member who did not develop the original model. Nagappan's thesis [9] describes the SIMULA model developed for protocol experimentation and eventual concept evaluation. Venkateshwaran's research [10] initiates with the *independent verification* of Nagappan's model (together with other team members) and includes an experimental assessment of dynamic reconfigurability using three network examples shown in Figure 3 [10, p. 83].

Three techniques of model verification are represented in the programmed model:

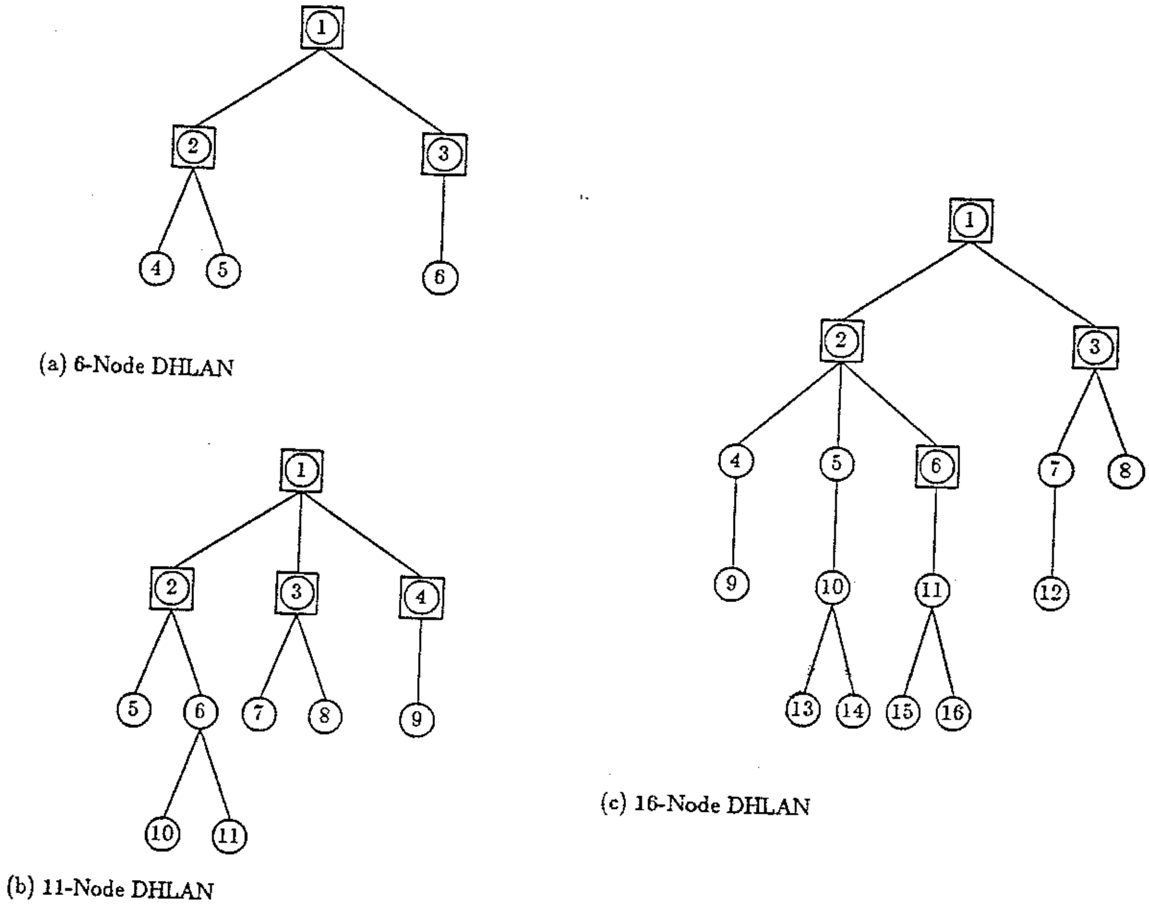
(1) *Desk checking*, consists of manual analysis of the model (flowcharts, diagrams, code) and cross-checking between model representations produced in prior phases of the model development.

(2) Trace checking of event occurrences to assure that messages follow the designated path, transmission delays are properly computed and accumulated, queues are correctly incremented and decremented, ACS messages are buffered during reconfiguration, apex reassignment follows the protocol definition, and the buffered messages are correctly depleted during the adjustment period.

(3) *Instrumentation-based testing,* with probes inserted at critical model points to assure that model behavior is reasonable and explainable; e.g., that messages are neither lost nor improperly delayed during reconfiguration or that model behavior follows expectations as message demands are increased or reduced.

Experimental Concept Evaluation

The three network examples in Figure 3 are the subjects by which comparisons are made between a static hierarchy (no reconfiguration) and the dynamic hierarchy. These comparisons are extracted from a carefully defined experiment design so that the results, considered in the context of the verification procedures, have assured credibility.



(a) 6-Node DHLAN

(b) 11-Node DHLAN

(c) 16-Node DHLAN

Figure 3. Network Examples for Evaluation

Experiment Design

The standard Simula random number generator for IBM installations is used throughout. This is a Lehmer multiplicative generator described in [14].

The method of batch means is used as the underlying basis for sample definition and subsequent analysis [15]. Two essential assumptions of the batch means method are tested:

(1) The normality of the distribution of sample means is examined with the UNIFIT package for distribution fitting [16], which clearly establishes the normal distribution as the best fit for the sample means produced from simulation of the 11-node network.

(2) Batch independence is established through the method proposed by Fishman [17, pp. 237-240], which identifies a sufficient batch size as 2000 messages. A plot of inter-batch correlation shows both positive and negative values with rapid damping to zero. The selected batch size of 30,000 messages represents a conservative decision.

Pilot runs for each network example are employed with a moving average technique to identify the conclusion of the transient period. That model requiring the largest number of messages before settling defines the termination of the transient period and the beginning or measurement for all examples [18]. Not unexpectedly, this value occurs for the 16-node example at slightly below 45,000 messages, which is used as the point for steady-state parameter estimation throughout the comparisons.

Basis for Comparison

In terms of improved survivability and reliability, the DHLAN clearly offers advantages. The dominant issue motivating this research is an assessment of the effect on average message transmission time in comparison with a static hierarchy. To gauge this effect, a constant message origination (demand) rate of 100 per second is imposed for each node. Destination nodes are selected according to five scenarios: heavy locality (destination probability varies proportionally with the distance), inverse locality, equal likelihood among destination nodes, heavy demand on ACS nodes, and high use of the apex. Reconfigurations can represent a transition from one scenario to another. However, space does not permit description of model results in terms of scenario interactions.

Two widely different message demand matrices are defined for each example network. Link capacity assignment for the static hierarchy, following the well-known square root assignment discipline, represents a compromise between the divergent demands. The square root assignment is followed for each of the demand conditions in the DHLAN, representing the capability for bandwidth reassignment.

The Comparative Results

Table 1 summarizes the results in terms of mean transmission delay for the three network examples according to the number of reconfigurations and the transition time (to effect each reconfiguration). The differences in mean transmission delay are almost imperceptible for the 6-node network yet dramatic for the larger two. An increased number of nodes forces greater congestion in the higher levels (unless the strong locality scenario is imposed) and the degradation in service is pronounced without dynamic adaptation.

Table 1. Comparison of Mean Message Transmission Delay for the Static and Dynamic Networks

Number of Nodes	External Demand Changes	Reconfig. Period (ms)	Average Delay (ms) (Confidence Interval Limits) Static	Dynamic
6	1	50	10.611 (10.453, 10.769)	9.304 (9.234, 9.374)
		300		9.992 (9.852, 10.131)
	3	50	10.468 (10.301, 10.634)	10.010 (9.904, 10.115)
		100		10.286 (10.087, 10.486)
11	1	50	105.907 (86.486, 125.328)	19.138 (18.765, 19.511)
		300		22.992 (22.429, 23.554)
	3	50	155.778 (130.022, 181.533)	18.068 (17.765, 18.371)
		100		19.099 (18.626, 19.571)
16	1	50	109.704 (27.946, 191.463)	20.932 (19.543, 22.320)
		300		24.195 (22.435, 25.955)
	3	50	264.221 (155.491, 372.951)	24.741 (23.168, 26.314)
		100		27.916 (26.333, 29.500)

The external demand changes on the static network cause an increase in delay from 47 percent (105.907 to 155.778) for the 11-node example to 141 percent (109.704 to 264.221) for the 16-node network. The comparable increase under the dynamic hierarchy, keeping the total time lost for transition fixed, ranges from -5 to 18 percent. At this point the decrease in mean delay experienced for the 11-node example is believed to result from a difference in the scenarios compared rather than a fundamental behavioral anomaly.

Confidence interval estimates obtained using the method of batch means for both the dynamic and static networks leave no doubt as to the significant reduction in delay realized with dynamic reconfigurability. Investigation is continuing as both scenario interactions and sensitivity issues are examined.

References

1. U. N. Bhat and R. E. Nance, " An Evaluation of CPU Efficiency Under Dynamic Quantum Allocation", *Journal ACM*, Vol. 26, No. 4, pp. 761-778, October 1979.
2. R. E. Nance, " Distributed Computing: An Overview", presentation at the Naval Surface Warfare Center, December 1979.
3. E. D. Jensen, " The Honeywell Experimental Distributed Processor -- An Overview", *Computer*, pp. 28-39, January 1978.
4. E. J. Derrick and R. E. Nance, " Local Area Networks and the Dynamic Hierarchy: A Tutorial", Tech. Rept. TR-86-18, Systems Research Center, Virginia Tech, March 1986.
5. R. E. Nance and R. L. Moose, Jr., " Link Capacity Assignment in Dynamic Hierarchical Networks", *Computer Networks and ISDN Systems*, to appear.
6. R. E. Nance, R. L. Moose, Jr., and R. V. Foutz, " A Statistical Technique for Comparing Heuristics: An Example from Capacity Assignment in Computer Networks", *Communications ACM*, Vol. 30, No. 5, pp. 430-442, May 1987.
7. E. D. Jensen, " The Honeywell Experimental Distributed Processor -- An Overview", *Computer*, pp. 28-39, January 1978.
8. R. L. Moose, Jr. " Modeling Networks with Dynamic Topologies", Tech. Rept. SRC-88-001, Systems Research Center, Virginia Tech, January 1988.
9. S. Nagappan, " Protocol Design and Analysis for a Dynamic Hierarchical Local Area Network", M.S. Thesis, Department of Computer Science, Virginia Tech,

June 1986.

10. A. Ventakeshwaran, " An Experimental Investigation of Dynamically Reconfigurable Computer Network Architectures Through Simulation", M.S. Thesis, Department of Computer Science, Virginia Tech, January 1988.

11. V. R. Dhadesugoor and P. V. Autenried, " Real-Time Simulation of a Prioritized Dynamic Access Protocol for Local Area Networks", Tech. Rept. CS 8607, Department of Computer Science, Stevens Institute of Technology, March 1986.

12. T. Rogeberg, " TETRASIM: A Program System for the Simulation of Telephone Networks", *Computer Networks and Simulation*, S. Schoemaker (ed.), North-Holland, Amsterdam, pp. 195-226, 1978.

13. P. J. Fortier and R. G. Leary, " A General Simulation Model for the Evaluation of Distributed Processing Systems", *Proceedings of the Annual Simulation Symposium*, pp. 215-226, 1980.

14. D. Hammarskjoldsvei, " SIMULA Programmer's Guide IBM 370", Simula a.s., N-0535, Oslo, 1986.

15. R. W. Conway, " Some Tactical Problems in Digital Simulation", *Management Science*, Vol. 10, No. 1, pp. 47-61, October 1963.

16. S. G. Vincent and A. M. Law, " UNIFIT: An Interactive Computer Package for Fitting Probability to Observed Data", User's Guide, Simulation Modeling and Analysis Company, Tucson, 1983.

17. G. S. Fishman, *Principles of Discrete Event Simulation*, John Wiley, New York, 1978.

18. J. R. Emshoff and R. L. Sisson, *Design and Use of Computer Simulation Models*, Macmillan, New York, 1970.

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Limited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) Systems Research Center Blacksburg, Virginia 24061 SRC-88-003	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Systems Research Center — 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION Naval Surface Warfare Center
6c. ADDRESS (City, State, and ZIP Code) Virginia Tech 320 Femoyer Hall Blacksburg, Virginia 24061	7b. ADDRESS (City, State, and ZIP Code) Dahlgren, Virginia 22448
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Naval Surface Warfare Center — 8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER N60921-83-G-A165-B032
8c. ADDRESS (City, State, and ZIP Code) Dahlgren, Virginia 22448	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

Dynamically Reconfigurable Networks: Concept Evaluation Through Simulation

12. PERSONAL AUTHOR(S)

A. Venkateshwaran, R.E. Nance, O. Balci

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim Research	FROM ______ TO ______	January 1988	15

16. SUPPLEMENTARY NOTATION

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The dynamically reconfigurable hierarchical network offers decided advantages in reliability and survivability. The concept is evaluated strictly in terms of message transmission delays for demand scenarios that vary widely. A SIMULA 67 model provides the experimental basis for obtaining preliminary results that show the reductions in transmission delays to be significant and to increase with the size of the network.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) — 22c. OFFICE SYMBOL