Technical Report TR-87-30*

Link Capacity Assignment in Dynamic Hierarchical Networks†

Richard E. Nance
Robert L. Moose, Jr.

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

July 1987‡

---

* Cross-referenced as SRC-87-011, Systems Research Center, Virginia Tech, Blacksburg, VA.

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

‡ This report supercedes the original report released May 1985, SRC-85-010 / TR-85-25.

Link Capacity Assignment in Dynamic Hierarchical Networks †

Richard E. Nance
Robert L. Moose, Jr.

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

Revised July 1987 ‡

† This research was supported in part by the Naval Sea Systems Command under Contract No. N60921-83-G-A165 through the Systems Research Center at Virginia Tech.
‡ Original release date: May 1985.

ABSTRACT

The dynamic hierarchy, which is a generalization of the centralized, tree structured network, is introduced in this paper. First, a queueing network model of the dynamic hierarchy is formulated. Following the derivation of a network performance measure, probabilistic and heuristic assignment strategies are created. These strategies are compared through the use of analysis of variance procedures and preferred strategies are selected. It is found that a relatively simple heuristic strategy produces capacity assignments whose quality is comparable to that of the preferred probabilistic strategy.

CR Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design — *distributed networks, network topology*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems; C.2.5 [**Computer-Communication Networks**]: Local Networks

General Terms: Design, Performance, Theory

Additional Key Words and Phrases: dynamic topology, link capacity assignment, reconfigurable systems

Link Capacity Assignment in Dynamic Hierarchical Networks

Richard E. Nance
Robert L. Moose, Jr.

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

1. Introduction

Two concepts dominate the current research in computer architecture: (1) the maximization of concurrency as a means for achieving increased computational speed ("power"), and (2) the incorporation of more sophisticated forms of adaptation ("reconfigurability") in the elements comprising a system. The first concept is popularly described as "distributed systems," and the second, "adaptive systems." The merger of both concepts occurs in research and development in "dynamically reconfigurable systems," in which the state transitions can be redefined in order to alter processing elements as well as the relationships among processing elements.

Concepts of dynamically adaptive behavior are advanced in research models of operating systems schedulers (for example, see [BLEP76], [BUNR72], [POTD76], and [BHAU79]). Techniques for diagnosis, self-repair and recovery have incorporated dynamically adaptive procedures (e.g., see [HAKS84] and [HOSS84]). An approach to the formal specification of dynamic distributed recovery is described by Yannez and Hayes [YANR86].

In general, the manifestation of distributed systems concepts in operational forms of local area networks (LANs) or wide area networks has underscored the need for more dynamic responsiveness to changes in an expanded control environment. One such example is the UniLAN, which automatically switches among three bus access techniques to adjust to demand and device loading [DAHA84]. Among wide area network applications, AT&T is touting the cost savings realized through the use of

Dynamic Nonhierarchical Routing (DNHR) in the Message Telecommunications Service (MTS) network [ASHG84]. The DNHR procedures, used by current 4ESS † switches, provides for alternative routing paths to be used during heavy demand periods during the day. Utilization of the alternative paths no longer follows the strictly hierarchical switch categories used in earlier architectures [BURP68]; DNHR is a "classless" switching network [MOCJ84, p. 24].

2. The Dynamic Hierarchy Concept

While dynamically adaptive behavior in LANs is often motivated by the goal of fault tolerance[YANR86, p. 871], this capability appears extremely attractive in applications where the external demands for service can experience sudden, significant shifts. The class of LANs under study in this work are embedded information transfer systems [MANJ74] where the encapsulating application system:

(1) requires real-time (or time-critical) response,

(2) experiences major variations in the demands placed on the embedded LAN (the message traffic and processing requirements), and

(3) imposes stringent requirements for high capability, reliability, survivability and adaptability that must be imparted as LAN requirements.

Military applications typify this class of LANs.

Within this class of LANs, the dynamic hierarchy is an architectural concept that:

(1) mandates an hierarchical topology (perhaps supporting a command hierarchy in the encapsulating system),

(2) permits the apex of the hierarchy, link capacities †, and other parameters to vary as either external or internal conditions change.

Network reconfiguration induces "costs" in two forms: (1) the overhead in space and time to enable dynamic change, and (2) the impairment of transmission service during the change. The benefits of

† 4ESS is a trademark of AT&T Bell Laboratories.

† Note that in general, references made here to variable link capacities should be interpreted as applying to dynamic hierarchy implementations based on a shared transmission medium such as a bus or ring. Only in a restrictive sense, when node pairs are connected by redundant (hot spare) links, may capacities in a point-to-point dynamic hierarchy be viewed as variable.

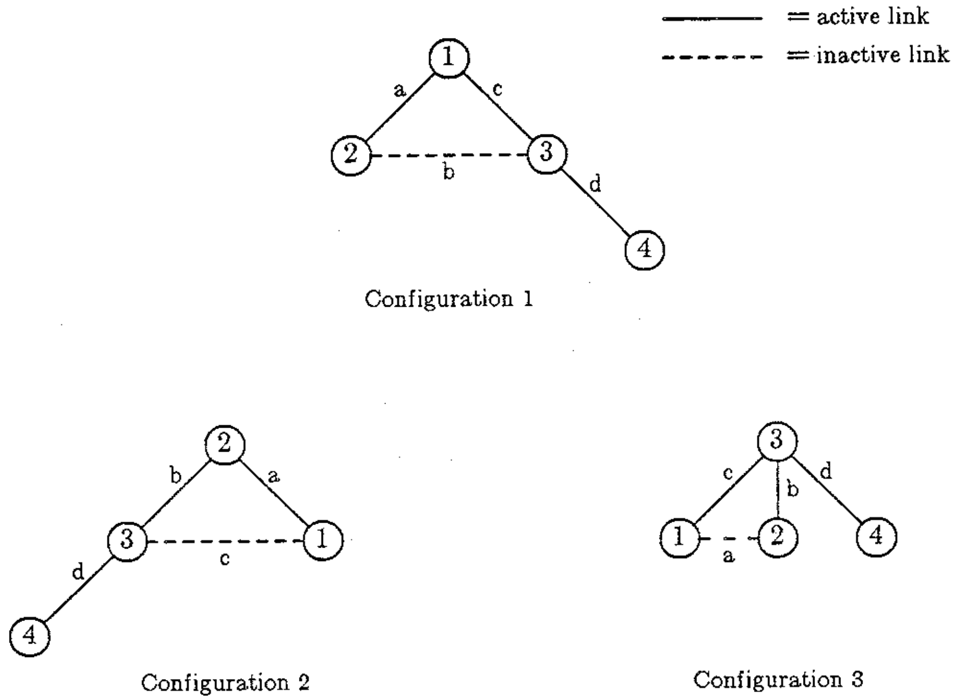
dynamic reconfigurability extend beyond the potential for tailoring service to demand. Increased reliability and survivability, improved security, and enhanced adaptability represent additional payoffs.

A *configuration* is defined by: (1) the designation of active nodes, (2) the designation of active links, which effectively establishes the connection bandwidth between node pairs, and (3) the designation of the apex node. The nodes comprising the apex candidate set (ACS) are fixed by the encapsulating system, but members could fail and become inactive. A *physical topology* defines the connectivity among the entire set of nodes for a given configuration. The *logical topology* is defined by enumeration of the physical topologies achievable within all configurations. Figure 1 illustrates the relationship between physical and logical topologies for a simple network of four nodes and four links.

In general, a dynamic hierarchy with K nodes has $L \geq K-1$ physical interconnections (links). In each configuration, there exist $K-1$ active links (whose traffic rates are nonzero) and $L-K+1$ inactive links (which carry no messages and thus have traffic rates of zero). The set of active links is such that the interconnections form a tree-structured topology. Equivalently, the network consisting of these $K-1$ links and the K nodes is connected. A link is inaccessible and may be considered nonexistent in any configuration in which it is inactive.

Different configurations generally have different sets of active and inactive links. A link may be active in every configuration, but no link may be inactive in every configuration. Note that in the case where $L = K-1$, the network topology is physically static. Every link is active in all configurations. Although the interconnections remain fixed, the network topology is logically variable as a results of changes in the apex node (and the corresponding changes in the hierarchical distribution of control responsibilities).

More subtle distinctions of logical and physical interconnections can be made; for example, the differences in channel bandwidth afforded by a bus controller employing a priority polling protocol. However, discussion of these issues is beyond the scope of this work.



(a) The Pysical Topologies Realizable in Each Configuration

Nodes (columns) × *Nodes* (rows)

	1	2	3	4
1		a	b	
2	a		c	
3	b	c		d
4			d	

(b) The Logical Topology in a Matrix Representation

Figure 1. Illustration of the Physical and Logical Topologies

3. Evaluation of the Concept

As a first approximation of the cost/benefit tradeoff realized from the dynamic hierarchy architecture, only the capability (service) potential is examined in a comparison of a static (conventional) tree-structured topology with a dynamic counterpart. An additional goal of this study is to investigate the applicability of analytical modeling approaches in lieu of total dependence on the more costly simulation models typically used. The following brief descriptions of the prototype model of a dynamic hierarchy and the analytical approach being investigated provide a context for the study of link capacity assignment, which is the focus of this paper.

3.1. Prototype Models of Dynamic Reconfigurability

The initial effort in concept evaluation employs simplifying assumptions regarding the forms of reconfigurability. While the nature of the relationship between the encapsulating system and the embedded LAN might appear restrictive, the characteristics are descriptive of a particular application domain: naval surface combat systems.

The essential characteristics of the model are:

(1) a packet-switched, strictly hierarchical network [NANR72, p. 242] in which the apex node can vary among a designated subset (the apex candidate set (ACS)),

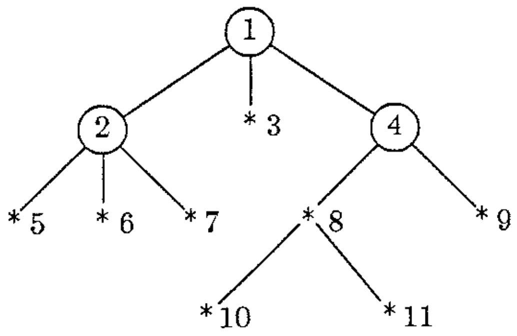
(2) the ACS and the specific topology, including the apex assignment, are designated by the encapsulating system for each environment (external or internal situation) encountered by that system,

(3) network behavior can be partitioned into three time periods: regular operation, reconfiguration (topological change being effected), and adjustment (packets buffered during reconfiguration are transmitted).

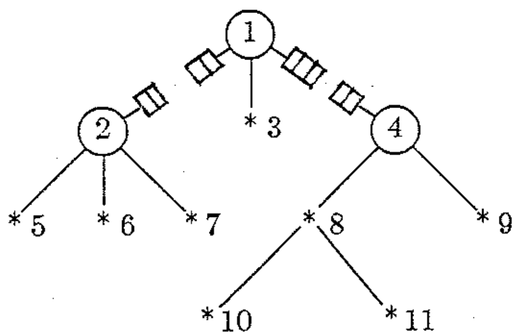
The cyclic transition induced by reconfiguration is illustrated in Figure 2. Protocol specifics are described in [NAGS86] and [BHAU87]. The latter paper describes a queueing network model using an approach entirely different from that which follows.

3.2. A Computational Queueing Model of the Dynamic Hierarchy Prototype

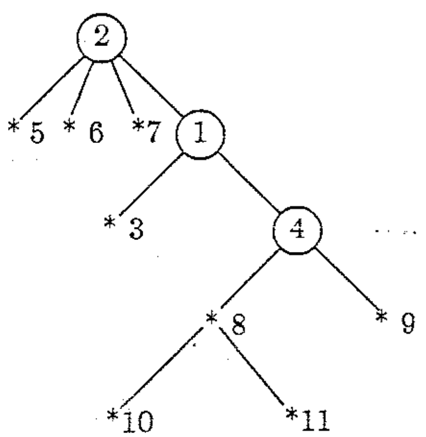
A simulation model addressing some of the capabilities for dynamic reconfigurability is intended for detailed examination of protocol correctness and performance comparisons. The cost and effort incurred by simulation inhibits the broadly based evaluation preferable for comparing the static and dynamic hierarchical organizations. Thus, a computational



(a) *Regular Operation*: Node 1 as apex and nodes 2 and 4 being the other members of the apex candidate set (ACS). The need for reconfiguration is triggered by a "broadcast" from an ACS member.



(b) *Reconfiguration*: Each ACS node buffers all message packets to other ACS members but continues to serve its local hierarchy. Reconfiguration can be aborted by the incumbent apex until the final apex transition is signaled.



(c) *Adjustment*: Completion of reconfiguration initiates an adjustment period during which the messages buffered during the transition are transmitted. When all messages interrupted by reconfiguration are transmitted, a period of regular operation begins.

Figure 2. The Periods of Operation Induced by Reconfiguration

model based on a network of queues in a random environment (*RE-network*) is under development. The term "computational" is intended to infer that the complexity of the queueing network permits only an approximate solution.

The proposed queueing model of the dynamic hierarchy represents a network generalization of what has become known as an $M/M/1$ queue in a random environment, (which is referred to here as an *RE-queue*). The RE-queue and related systems are discussed extensively in the literature. Early work includes that of Eisen and Tainiter [EISM63], Scott [SCOM64], Neuts [NEUM69], Yechiali and Naor [YECU71,YECU73]. More recent work includes that of Neuts [NEUM77,NEUM78,NEUM81], Purdue [PURP74,PURP78], and others.

The approach adopted in this work is similar to early studies[MEIB72], which utilize a separation of link and nodal service functions. The model that follows is viewed as a first step toward a more comprehensive analytical treatment that joins the two forms of service. Despite the separation strategy, the capacity assignment analysis for the dynamic hierarchy introduces sufficient complexity to force approximate solution techniques.

The prominent assumptions underlying this queueing network model include the following (This material is taken from [MOOR87]):

(1) External situation changes occur according to a random (Markovian) environment process.

(2) When the environment process is in state i, messages with source node j and destination node k arrive at node j according to a Poisson process with rate $\gamma_{jk}^{(i)}$.

(3) Given the state of the environment process, the arrival processes of (2) are mutually independent.

(4) Messages pass through the network in a store-and-forward fashion.

(5) At each node, there exists a separate FCFS queue with infinite waiting room for each outgoing link.

(6) The lengths of arriving messages are exponentially distributed with mean μ^{-1}. Additionally, at each intermediate node in the path of a message, the length of the message is reset according to the same exponential distribution (following

Kleinrock's independence assumption [KLEL64, pp. 49-50]; see also [KLEL76]).

(7) The message length processes of (6) are mutually independent.

(8) The collections of arrival and message length processes are mutually independent.

Under these assumptions, definition of the appropriate stochastic process representing a network of $M/M/1$ queues in a random environment is straightforward. This process forms the basic model of the dynamic hierarchy.

Let $E = \{E(t);\ t \geq 0\}$, the environment process, be a stationary, irreducible Markov process with finite state space $S_E = \{1, 2, \ldots, M\}$, where M is the number of possible environments. Consider a network of L queues, each of which has infinite waiting room and serves customers according to a FCFS discipline. Each queue $l \in \{1, 2, \ldots, L\}$ operates under the influence of the environment process as follows: On $\{E(t) = i\}$, $i \in S_E$

(1) Arrivals from outside the network (external arrivals) occur according to a Poisson process with rate $\lambda_l^{(i)}$.

(2) Service times are independent, identically distributed random variables following the exponential distribution with mean μ^{-1}.

Assume that the service time processes are mutually independent, given the state of E, the external arrival processes are mutually independent, and the collections of service time processes and arrival processes are independent.

Now let $\underset{\sim}{N} = \{\underset{\sim}{N}(t);\ t \geq 0\}$ be the process of network queue lengths, where

$$\underset{\sim}{N}(t) = (N_1(t), N_2(t), \ldots, N_L(t))$$

and $N_l(t)$ is the length (including the customer in service, if any) of queue l at time t. Then the process of interest, $(E, \underset{\sim}{N}) = \{E(t), \underset{\sim}{N}(t);\ t \geq 0\}$ is a Markov process with state space

$S_E \times (\bigtimes_{l=1}^{L} S_{N_l})$, where $S_{N_l} = \{0, 1, 2, \cdots\}$ is the state space of a single queue length process.

To model the dynamic hierarchy as described above requires specification of the correspondence between the external arrival processes in source/destination node pair form and the resultant composite arrival processes to the individual link servers. Each composite arrival process is the superposition of a number of external arrival processes and a number of departure processes from other links. Further exact analysis of the RE-network requires (at least) derivation of the rates of these departure processes, a task that is beyond the scope of this research (The composite processes of external arrivals to a link are superpositions of independent Poisson processes and thus Poisson[CINE75]).

4. Performance and Capacity Assignment

Due to complicating factors, the analysis carried out thus far is highly approximate in nature. First, an approximation of mean network delay (sojourn time) is derived. This approximation represents the basic measure of network performance for subsequent optimization and analysis. Next, a number of suboptimal, probabilistic and heuristic capacity assignment strategies are defined. Statistical techniques are then used to compare the effects of these strategies and to select the best strategy or strategies. Further details of this work are found in [MOOR83].

4.1. Performance measure

Mean network delay is taken as the primary measure of performance of the dynamic hierarchy. For conventional networks, given the assumptions of Kleinrock [KLEL64,KLEL76], a closed form expression for mean delay is derived through the application of elementary queueing theory. This expression is extended as follows to provide an approximate measure of delay in the dynamic hierarchy.

Define

$T^{(i)} =$ mean delay of messages in configuration i,

$T_j =$ mean delay on link j across all configurations, and

$\xi^{(i)} =$ stationary probability of occurrence of configuration (and environment) i †.

Then, considering each configuration i as if it were the topology of a static network, let $T^{(i)}$ be mean delay as derived by Kleinrock [KLEL76, pp. 314-317, 320-322].

Then, we take the measure of network delay for the dynamic hierarchy to be weighted sum of the individual configuration delays. That is,

$$T = \sum_{i=1}^{M} \xi^{(i)} T^{(i)}$$

This expression for T has the intuitively appealing characteristics that

(1) The contribution to the total delay made by each configuration is related to its stationary probability of occurrence.

(2) When $M = 1$ ($\xi^{(1)} = 1.0$) it reduces to the formula for mean delay in a conventional network.

The expression appears, in general, to be a conservative estimate of delay, often predicting infinite delay when this is probably not the case. Despite this defect, the estimate proves useful as a delay measure for the purpose of comparing capacity assignment strategies within the class of dynamic hierarchical networks.

4.2. Capacity assignment strategies

The ideal goal is to solve the following optimization problem:

Given

(1) the set of network configurations and

(2) for each configuration (environment) i, its

(a) stationary probability of occurrence $\xi^{(i)}$ and

(b) traffic matrix $\Gamma^{(i)}$,

† These probabilities are nonzero since the state space of the environment process is irreducible and finite.

$$\text{minimize } C = \sum_{j=1}^{L} C_j$$

with respect to $\{C_j\colon j = 1, 2, \ldots, L\}$

$$\text{subject to } T = \sum_{i=1}^{M} \xi^{(i)} T^{(i)} \leq T_{MAX},$$

where C is the total cost (capacity †) and T_{MAX} is the upper bound on mean network delay.

4.2.1. Probabilistic strategies

Although the resultant capacity assignment strategies are not optimal, the alternate approach described in this section eliminates the need to solve a complex queueing model. The approach allows the development of probabilistic assignment strategies through generalization of previous analytic. The method of definition is heuristic in that it uses a simple rule of thumb to adapt proven, conventional network formulae for use in the dynamic hierarchy. It is straightforward and conforms to the idea on which the dynamic hierarchy delay measure is based.

Consider the single configuration i in a dynamic hierarchy. For the moment, consider this configuration to represent a static hierarchy. Past research derives some function f_j, through which optimal capacities are assigned. In our case, f_j depends on the parameters corresponding to configuration i.

In applying this to the dynamic hierarchy, first recall that the collection $\{\xi^{(i)}\colon i = 1, 2, \ldots, M\}$ of configuration probabilities is given. Then, for the capacity assigned to link j, set

$$C_j = \max_{1 \leq i \leq M} \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \sum_{i=1}^{M} \xi^{(i)} f_j'(\text{configuration } i \text{ parameters}), \; j = 1, 2, \ldots, L, \qquad (1)$$

† As reflected in the objective function, a linear, unit (continuous) cost function is assumed so that the cost of x units of capacity is equal to x.

where the function f_j' is related to f_j as illustrated in the following example.

As an example of this construction method consider the dual of Kleinrock's square root assignment strategy [KLEL76, pp. 329-332,350]. For conventional networks, a simplified form of this strategy is defined as follows:

$$C_j = f_j = \frac{\lambda_j}{\mu} + \frac{\sqrt{\lambda_j}}{\mu\gamma T_{MAX}} \sum_{l=1}^{L} \sqrt{\lambda_l},\ j = 1, 2, \ldots, L.$$

To adapt this strategy for application to the dynamic hierarchy, let

$$f_j'(\text{configuration } i \text{ parameters}) = \frac{\sqrt{\lambda_j^{(i)}}}{\mu\gamma^{(i)} T_{MAX}} \sum_{l=1}^{L} \sqrt{\lambda_l^{(i)}}$$

so that (1) becomes

$$C_j = \max_{1 \le i \le M} \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \sum_{i=1}^{M} \xi^{(i)} \frac{\sqrt{\lambda_j^{(i)}}}{\mu\gamma^{(i)} T_{MAX}} \sum_{l=1}^{L} \sqrt{\lambda_l^{(i)}},\ j = 1, 2, \ldots, L.$$

In this assignment, the first term represents the minimum required capacity. Examination of the proposed delay measure reveals that for each link j, unless $C_j > \frac{\lambda_j^{(i)}}{\mu}$ for all i, the link experiences unbounded delay, causing unbounded delay for the network as a whole.

Alternately, a minmax basis, which might be more appropriate in the in the presence of time-critical delay constraints, may be employed. The strategy DMX given in Section A.2 is based the solution of the dual of Meister's optimization objective [MEIB71] with $k \to \infty$, which represents a minmax criterion. The remaining strategies are listed in Appendix A (A.1: primary problem— delay minimization, A.2: dual problem— cost minimization).

4.2.2. Heuristic strategies

The heuristic capacity assignment strategies differ from the probabilistic strategies in two respects. First, the heuristic strategies are defined algorithmically rather than being determined as an extension of of a Lagrangian optimization result. Second, a unit cost function is assumed, but a discrete cost structure is allowed.

The method of Maruyama, et al. (see [MARK76], for example) is used to construct the heuristic strategies. That is, first a collection of capacity assignment heuristics are defined. These heuristics are: SETLOW, SETHIGH, ADD1, ADD2, DROP1, DROP2, ADDALL, and DROPALL. Various combinations of these heuristics are then formed to produce a set of composite assignment strategies. Table 1 provides a breakdown of the strategies into their heuristic components. For each row k in this table, an i in column j means that heuristic j is the ith step of strategy k.

Table 1 Composition of Heuristic Strategies

	Heuristic							
Strategy	SETLOW	SETHIGH	ADD1	ADD2	DROP1	DROP2	ADDALL	DROPALL
HUR.1	1		2					
HUR.2	1			2				
HUR.3		1			2			
HUR.4		1				2		
HUR.5	1		2		4		3	
HUR.6	1			2	4		3	
HUR.7	1		2			4	3	
HUR.8	1			2	4		3	
HUR.9		1	4		2			3
HUR.10		1	4			2		3
HUR.11		1		4	2			3
HUR.12		1		4		2		3

The capacity assignment heuristics and the heuristic strategies are described in detail in Appendix B.

Let $A = \{a_1, a_2, \ldots, a_H\}$ be the discrete set of available capacities. Assume that that for each $a_k \in A$ the cost of assigning a_k units of capacity to any link j is $d_j(a_k) = a_k$. An additional assumption is that capacities are available in increments of a given constant and up to a given maximum. The set of available capacities used in the heuristic strategy experiments is $A = \{128, 256, 384, \ldots, 12800\}$.

Each of the capacity assignment heuristic performs one of the following functions:

- Generation of an initial set of assignments.

- Choice of a link for a capacity increase.

- Choice of a link for a capacity decrease.

- Raise or lower all capacities.

Heuristic (1) sets C_j to the smallest available capacity greater than or equal to $\max_{1 \le i \le M} \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\}$, thus yielding capacities under which all delays are finite (according to the formulae for T_j and U_j). Heuristic (2) sets all capacities at the maximum available, a_H. This heuristic can be used as part of a feasibility test. After execution of (2), since all capacities are at the highest possible value, if all delays are finite, the problem is feasible.

Heuristics (3) and (4) increase a link capacity to the next highest available based on delay/cost ratios. Heuristic (3) uses T_j, the weighted link delay measure for the dynamic hierarchy. By weighting each $(\mu C_j - \lambda_j^{(i)})^{-1}$ term by $\frac{\lambda_j^{(i)}}{\gamma^{(i)}}$, it calculates a value which equals the contribution of link j to overall delay. Thus, (3) raises the capacity of that link whose increase yields the maximum decrease in network delay per unit of cost increase. The difference in (4) is that it uses unweighted link delay, which measures individual link delays rather than overall network delay. So, (4) raises the capacity of that link which experiences the maximum delay decrease per unit of cost increase.

Similarly, (5) and (6) produce capacity decreases. Heuristic (5) uses weighted delay and lowers the capacity of a link so that the network realizes the minimum delay increase per unit of cost decrease. Heuristic (6) uses unweighted delay.

The purpose of (7) is to raise the capacity of each link j for which $C_j < a_H$. Similarly, (8) lowers the capacities of all links for which $C_j > a_1$. These are used as interfaces in combining various of the other heuristics.

Each combination of the capacity assignment heuristics corresponds to a different strategy. Two strategy types exist:

- Those which use one of SETLOW and SETHIGH and one of ADD1, ADD2, DROP1, and DROP2, and
- Those which use one of SETLOW and SETHIGH, two of ADD1, ADD2, DROP1, and DROP2, and one of ADDALL and DROPALL.

These strategies are defined in Section B.2 Those in the first group each contain a single iterative step. First they generate a starting point. They then perform the iterative step, increasing or decreasing capacities, until they reach a stopping point.

Strategies of type (A) in Appendix B.2 start at the bottom and work up to a point where the delay constraint is satisfied. Strategies of type (B) start at the top and decrease capacities until the constraint is violated or all capacities are at a_1. If the delay bound is exceeded, capacity is restored to the link which is the victim of the last drop (and thus causes the violation of the constraint). Note that the given blocks assume feasibility.

Members of the second group use two iterative steps. After the generation of initial capacities, they execute the first set of iterations to produce a first stopping point. Then one of the interfaces ADDALL and DROPALL is applied to set capacities at one step past this stopping point. If the strategy uses an ADD in its first iteration, the interface is ADDALL. Otherwise, it is DROPALL. Given this new set of assignments the strategies next execute a second iterative step. If the first step increases capacities, the second decreases capacities and vice versa.

Note that the heuristic strategies **are** tailored for use in designing dynamic hierarchical networks. The superscript (i) in the definitions of C_j, T_j, and U_j and most of the heuristics denotes a dependence on a varying environment and configuration. Static networks do not exhibit this type of variablility and thus such a dependence is not present in strategies for their design.

5. Results

Having developed the two sets of strategies, the final necessary steps are to compare the strategy and identify the preferred strategies. Due to the approximate and heuristic nature of the strategies, exact, analytic comparisons are not possible. Further, extension of the results to all dynamic hierarchies is required. These considerations lead to the use of statistical comparison methods, namely analysis of variance (ANOVA) procedures. Use of ANOVA procedures for this purpose is described in detail in[NANR87]. The remainder of this section provides an overview of that work.

The approach consists of first performing capacity assignments for a collection of test networks together with various sets of network parameters (*auxiliary parameter settings* or *a-settings*). Then performance measures under each set of capacity assignments are generated. If the test networks and a-settings are considered representative of the populations of dynamic hierarchical networks and a-settings, respectively, then the performance measures resulting from different capacity assignment strategies may be treated as random variables.

Taking this view, the experimental design is a nested, three factor design with fixed and random effects. Letting total network cost be the performance measure of interest †, the statistical model is

$$X_{ijk} = \mu + N_i + P_{(i)j} + S_k + (NS)_{ik} + (PS)_{(i)jk},$$

where

$X_{ijk} =$ total cost resulting from the application of strategy k to network i and its jth a-setting,

$\mu =$ mean of parent population,

$N_i =$ main effect of network i,

† It is assumed that the optimization objective is to minimize total cost subject to a constraint on mean network delay so that network cost is the measure on which the comparisons are based.

$P_{(i)j} =$ main effect of a-setting j within network i,

$S_k =$ main effect of strategy k,

$(NS)_{ik} =$ interaction effect of network i and strategy k, and

$(PS)_{(i)jk} =$ interaction effect of a-setting $(i)j$ and strategy k.

The assumptions underlying this model are discussed in[NANR87, pp. 433-434].

To compare the strategies HEURISTIC1, HEURISTIC2, ..., HEURISTIC12, X_{ijk} is (logarithm of) total cost for network, a-setting, strategy ijk and S_k is the effect of strategy k (HEURISTICk) on total cost. Additionally, let $\sigma^2_{(NS)}$ be the variance of network/strategy interaction effects. The test for strategy effects is

(1) H_0: $S_1 = S_2 = \cdots = S_{12} = 0$

H_a: $S_k \neq 0$ for at least one k.

The test for network/strategy interaction is

(2) H_0: $\sigma_{(NS)} = 0$

H_a: $\sigma_{(NS)} > 0$.

The experimental results and ANOVA computations yield the values given in Table 2.

Table 2. Analysis of variance (total cost) †

Source of Variance	Sum of Squares	Degrees of Freedom	Mean Square
Networks	101.3543	5	20.2709
A-Settings	319.4691	48	6.6556
Strategies	99.1074	11	9.0098
Network/Strategy Interaction	46.5195	55	0.8458
A-Setting/Strategy Interaction	37.1555	528	0.0704
Total	603.6059	647	0.9329

Summarizing the results of the two tests: in each case H_0 is rejected. We conclude that strategy effects are different (from (1)) but that differences in total cost are partially attri-

† Reprinted from [NANR87] (Table I) with permission of the ACM.

butable to network/strategy interaction (from (2)).

Following these tests and conclusions, multiple comparisons are performed to characterize further the differences among the strategies. Five additional sets of ANOVA computations and pairs of tests are performed bases on the following groups:

- {1, 2, 3, 5, 6, 7, 8, 9, 11}
- {1, 2, 3, 5, 6, 7, 8, 9, 11, 12}
- {1, 2, 3, 5, 6, 7, 8, 9, 10, 11}
- {1, 2, 3, 4, 5, 6, 7, 8, 9, 11}
- {4, 10, 12}

The results of these tests are listed in Table 3.

Table 3. Results of multiple comparisons †

	Strategy			Network/Strategy Interaction		
Test	Test Statistic ‡ and Value	Critical Value	Reject H_0	Test Statistic ‡ and Value	Critical Value	Reject H_0
A	$F_S = \frac{M_S^A}{M_{(NS)}^O} = 0.0220$	$F_{0.05,8,55} = 2.12$	no	$F_{(NS)} = \frac{MS_{(NS)}^A}{M_{(PS)}^O} = 0.0227$	$F_{0.05,40,528} \doteq 1.4725$	no
B	$F_S = \frac{M_S^B}{M_{(NS)}^O} = 5.1571$	$F_{0.05,9,55} = 2.06$	yes	$F_{(NS)} = \frac{MS_{(NS)}^B}{M_{(PS)}^O} = 5.9233$	$F_{0.05,45,528} \doteq 1.455$	yes
C	$F_S = \frac{M_S^C}{M_{(NS)}^O} = 5.2140$	$F_{0.05,9,55} = 2.06$	yes	$F_{(NS)} = \frac{MS_{(NS)}^C}{M_{(PS)}^O} = 5.8631$	$F_{0.05,45,528} \doteq 1.455$	yes
D	$F_S = \frac{M_S^D}{M_{(NS)}^O} = 5.2523$	$F_{0.05,9,55} = 2.06$	yes	$F_{(NS)} = \frac{MS_{(NS)}^D}{M_{(PS)}^O} = 5.9247$	$F_{0.05,45,528} \doteq 1.455$	yes
E	$F_S = \frac{M_S^E}{M_{(NS)}^O} = 0.00041$	$F_{0.05,2,55} = 3.17$	no	$F_{(NS)} = \frac{MS_{(NS)}^E}{M_{(PS)}^O} = 0.00057$	$F_{0.05,10,528} \doteq 1.89$	no

We draw the overall conclusions that the heuristic strategies may be divided into two groups, with differences existing between groups but not within groups. Furthermore, the strategies in the first group {1, 2, 3, 5, 6, 7, 8, 9, 11} are preferred over those in the other

† Reprinted from [NANR87] (Table III) with permission of the ACM.
‡ $MS_{\_}^{i}$ = mean square statistic from ANOVA table for comparison i. $MS_{\_}^{O}$ = mean square statistic from Table 2.

group {4, 10, 12}.

6. Conclusions

The research described herein has served as an introduction to various problems associated with and the modeling and analysis of dynamic hierarchical networks. A hierarchical network of this type is dynamic in that during its operation the apex node and the sets of active and inactive links may vary, thus inducing a reconfigurable, adaptive topology.

A number of advantages of the dynamic hierarchy over conventional networks are conjectured in this paper. Chief among these advantages is the ability to more easily satisfy timing constraints imposed by a time-critical application. Others include increases in reliability and survivability.

This research effort has not yet reached the point where such advantages are quantitatively demonstrable. Rather, work carried out thus far has concentrated on design and analysis *within* the class of dynamic hierarchical networks. The following tasks have been accomplished:

(1) Definition of a queueing model of the dynamic hierarchy and derivation of an approximate measure of network delay.

(2) Creation of probabilistic capacity assignment strategies.

(3) Definition of a set of capacity assignment heuristics and algorithmic combination of various heuristics to produce heuristic capacity assignment strategies.

(4) Comparison of the strategies resulting from (3) and selection of preferred heuristic strategies,

Task (4) is accomplished through the use of statistical comparison (analysis of variance) techniques and is dealt with with in the work previously cited.

Substantiation of the previously mentioned conjectures has not yet been attempted. To do so requires one or more performance measures for the dynamic hierarchy that are (theoretically) exact or are approximate but of quantifiable distance from the exact values.

Such measures would play a central role in comparing the performance of the dynamic hierarchy with that of conventional networks. This remains a primary goal of future research.

Acknowledgements

Tables 2 and 3 are from Nance, Moose, and Foutz [NANR87], Copyright 1987, Association for Computing Machinery, Inc., reprinted by permission. Karen Kaster deserves thanks for assisting with the typing and editing of this paper.

Appendix A: Probabilistic Assignment Strategies

This appendix contains the formulae for the probabilistic strategies. It is divided into Sections A.1: Delay Minimization Strategies, which attempt to minimize mean network delay subject to a constraint on total cost (the primary problem) and A.2: Cost Minimization Strategies, which attempt to minimize cost subject to a constraint on mean network delay (the dual problem). These strategies are named according to the strategies for conventional networks on which they are based.

A.1: Delay Minimization Strategies

(1) Proportional

$$C_j = \max_i \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \sum_{i=1}^{M} \xi^{(i)} \left[C - \sum_{l=1}^{L} \max_h \left\{ \frac{\lambda_l^{(h)}}{\mu} \right\} \right] \frac{\lambda_j^{(i)}}{\lambda^{(i)}} = \max_i \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \left[C - \sum_{l=1}^{L} \max_h \left\{ \frac{\lambda_l^{(h)}}{\mu} \right\} \right] \sum_{i=1}^{M} \xi^{(i)} \frac{\lambda_j^{(i)}}{\lambda^{(i)}}$$

(2) Square Root

$$C_j = \max_i \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \sum_{i=1}^{M} \xi^{(i)} \left[C - \sum_{l=1}^{L} \max_h \left\{ \frac{\lambda_l^{(h)}}{\mu} \right\} \right] \frac{\sqrt{\lambda_j^{(i)}}}{\sum_{l=1}^{L} \sqrt{\lambda_l^{(i)}}} = \max_i \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \left[C - \sum_{l=1}^{L} \max_h \left\{ \frac{\lambda_l^{(h)}}{\mu} \right\} \right] \sum_{i=1}^{M} \xi^{(i)} \frac{\sqrt{\lambda_j^{(i)}}}{\sum_{l=1}^{L} \sqrt{\lambda_l^{(i)}}}$$

(3) Min max

$$C_j = \max_i \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \sum_{i=1}^{M} \xi^{(i)} \left[C - \sum_{l=1}^{L} \max_h \left\{ \frac{\lambda_l^{(h)}}{\mu} \right\} \right] \frac{1}{L} = \max_i \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \left[C - \sum_{l=1}^{L} \max_h \left\{ \frac{\lambda_l^{(h)}}{\mu} \right\} \right] \frac{1}{L}$$

A.2: Cost Minimization Strategies

(4) Dual square root (DSR)

$$C_j = \max_i \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \sum_{i=1}^{M} \xi^{(i)} \sqrt{\lambda_j^{(i)}} \frac{\sum_{l=1}^{L} \sqrt{\lambda_l^{(i)}}}{\mu \gamma^{(i)} T_{MAX}}$$

(5) Dual min max (DMX)

$$C_j = \max_i \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \frac{1}{\mu T_{MAX}}$$

(6) Dual rth root

$$C_j = \max_i \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\} + \sum_{i=1}^{M} \xi^{(i)} \frac{\lambda_j^{(i)1/(r+1)}}{\mu T_{MAX}} \left(\sum_{l=1}^{L} \frac{\lambda_l^{(i)1/(r+1)}}{\gamma^{(i)}} \right)^{1/r}$$

Appendix B: Heuristic Assignment Strategies

The heuristic strategies are defined by first defining a collection of simple capacity assignment heuristics, which initialize or change capacities, and the combining these heuristics algorithmically. These heuristics and strategies use a discrete cost structure. Before describing these heuristics, we introduce the following notation and expressions. For each link $j \in \{1, 2, \ldots, L\}$ let

$C_j =$ the current capacity assignment of link j,

$T_j =$ the *weighted delay measure* for link j,

$$= \sum_{i=1}^{M} \xi^{(i)} \frac{\lambda_j^{(i)}}{\gamma^{(i)}} \frac{1}{\mu C_j - \lambda_j^{(i)}},$$

$U_j =$ the *unweighted delay measure* for link j,

$$= \sum_{i=1}^{M} \xi^{(i)} \frac{1}{\mu C_j - \lambda_j^{(i)}}.$$

B.1: Capacity assignment heuristics

The following is a list of the dynamic hierarchy capacity assignment heuristics (Note that j and k, link numbers, are taken to be elements of $\{1, 2, \ldots, L\}$ in all of these):

(1) SETLOW

For each j

Compute $l_j = \max_{1 \le i \le M} \left\{ \frac{\lambda_j^{(i)}}{\mu} \right\}$;

Find the smallest $a' \in A$ such that $L_j \geq a'$;

Set $C_j = a'$.

(2) SETHIGH

For each j

Set $C_j = a_H$.

(3) ADD1

For each j, let

$C_j' =$ the next capacity in A greater than C_j,

$$T_j' = \sum_{i=1}^{M} \xi^{(i)} \frac{\lambda_j^{(i)}}{\gamma^{(i)}} \frac{1}{\mu C_j' - \lambda_j^{(i)}};$$

Find k such that

$$\frac{T_k - T_k'}{C_k' - C_k} = \max_{1 \le j \le L} \frac{T_j - T_j'}{C_j' - C_j};$$

Set $C_k = C_k'$.

(4) ADD2

For each j, let

$C_j' =$ the next capacity in A greater than C_j,

$$U_j' = \sum_{i=1}^{M} \xi^{(i)} \frac{1}{\mu C_j' - \lambda_j^{(i)}};$$

Find k such that

$$\frac{U_k - U_k'}{C_k' - C_k} = \max_{1 \le j \le L} \frac{U_j - U_j'}{C_j' - C_j};$$

Set $C_k = C_k'$.

(5) DROP1

For each j, let

$C_j'' =$ the next capacity in A less than C_j;

$$T_j'' = \sum_{i=1}^{M} \xi^{(i)} \frac{\lambda_j^{(i)}}{\gamma^{(i)}} \frac{1}{\mu C_j'' - \lambda_j^{(i)}};$$

Find k such that

$$\frac{T_k'' - T_k}{C_k - C_k''} = \max_{1 \le j \le L} \frac{T_j'' - T_j}{C_j - C_j''};$$

Set $C_k = C_k''$.

(6) DROP2

For each j, let

$C_j'' =$ the next capacity in A less than C_j;

$$U_j'' = \sum_{i=1}^{M} \xi^{(i)} \frac{1}{\mu C_j'' - \lambda_j^{(i)}};$$

Find k such that

$$\frac{U_k'' - U_k}{C_k - C_k''} = \max_{1 \le j \le L} \frac{U_j'' - U_j}{C_j - C_j''};$$

Set $C_k = C_k''$.

(7) ADDALL

For each j such that $C_j < a_H$,

Set $C_j = C_j'$ (= the next capacity in A greater than C_j).

(8) DROPALL

For each j such that $C_j > a_1$,

Set $C_j = C_j''$ (= the next capacity in A less than C_j).

B.2: Heuristic strategies

The following blocks define the four members of of the first group of heuristic strategies:

(A) SETLOW;

WHILE ($T > T_{MAX}$) DO

ADD1 or ADD2;

(B) SETHIGH;

REPEAT

DROP1 or DROP2;

{$i =$ link whose capacity was just dropped}

UNTIL ($T > T_{MAX}$) or ($C_j = a_1$ for all j);

IF ($T > T_{MAX}$)

THEN increase the capacity of link i;

The second group of heuristic strategies are defined as follows:

(C) SETLOW;

WHILE ($T > T_{MAX}$) DO

ADD1 or ADD2;

ADDALL;

REPEAT

DROP1 or DROP2;

{$i =$ link whose capacity was just dropped}

UNTIL ($T > T_{MAX}$) OR ($C_j = a_1$ for all j);

IF ($T > T_{MAX}$)

THEN increase the capacity of link i;

```
(D) SETHIGH;
    REPEAT
        DROP1 or DROP2;
        {i = link whose capacity was just dropped}
    UNTIL (T > T_MAX) OR (C_j = a_1 for all j);
    IF (T > T_MAX)
        THEN increase the capacity of link i;
    DROPALL;
    WHILE (T > T_MAX) DO
        ADD1 or ADD2;
```

REFERENCES

ASHG84. G. R. Ash and V. S. Mummert, "AT&T Carves New Routes in Its Nationwide Network," *AT&T Bell Laboratories Record* **62**(6), pp. 18-22 (August 1984).

BHAU79. U. N. Bhat and R. E. Nance, "An Evaluation of CPU Efficiency Under Dynamic Quantum Allocation," *Journal of the ACM* **26** (4), pp. 761-778 (October 1979).

BHAU87. U. N. Bhat and R. E. Nance, "A Queueing Network Analysis of Dynamic Reconfigurability in an Hierarchical Information Network," Technical Report SRC 86-007, Systems Research Center, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (March 1987).

BLEP76. P. R. Blevins and C. V. Ramamoorthy, "Aspects of a Dynamically Adaptive Operating System," *IEEE Transactions on Computers* **C-25**(7), pp. 713-725 (July 1976).

BUNR72. R. B. Bunt and J. N. P. Hume, "Self Regulating Operating Systems," *INFOR* **10**(3), pp. 232-239 (October 1972).

BURP68. P. J. Burke, "Automatic Overload Controls in a Circuit-Switched Communications Network," *Proceedings of the National Electronics Conference*, pp. 667-672 (December 1968).

CINE75. E. Cinlar, *Stochastic Processes,* Prentice-Hall , Englewood Cliffs (1975).

DAHA84. A. M. Dahod, "Local Network Responds to Changing System Needs," *Computer Design* **23**(6), pp. 61-67 (June 1984).

EISM63. M. Eisen and M. Tainter, "Stochastic Variations in Queueing Processes," *Operations Research* **11**(6), pp. 922-927 (November-December 1963).

HAKS84. S. L. Hakimi and K. Nakajima, "On Adaptive System Diagnosis," *IEEE Transactions on Computers* **C-33**(3), pp. 234-240 (March 1984).

HOSS84. S. H. Hosseini, J. G. Kuhl, and S. M. Reddy, "A Diagnosis Algorithm for Distributed Computing Systems With Dynamic Failure and Repair," *IEEE Transactions on Computers* **C-33**(3), pp. 223-233 (March 1984).

KLEL64. L. Kleinrock, *Communication Nets Stochastic Message Flow and Delay,* McGraw-Hill, New York (1964).

KLEL76. L. Kleinrock, *Queuing Systems Volume II: Computer Applictions,* John Wiley & Sons, New York (1976).

MANJ74. J. H. Marley, "Embedded Computers-- Software Cost Considerations," *Proc. National Computer Conf.*, pp. 343-347 (May 6-10, 1974).

MARK76. K. Maruyama and D. T. Tang, "Discrete Link Capacity Assignment in Communication Networks," *Proc. 3rd International Conf. on Computer Communication,* pp. 92-97 (August 1976).

MEIB71. B. Meister, H. R. Mueller, and H. R. Rudin, Jr., "New Optimization Criteria for Message-Switching Networks," *IEEE Transactions on Communications* **COM-19**(3), pp. 256-260 (June 1971).

MEIB72. B. Meister, H. R. Muller, and H. R. Rudin, Jr., "On the Optimization of Message-Switching Networks," *IEEE Transactions on Communications* **COM-20**(1), pp. 8-14 (February 1972).

MOCJ84. J. M. Mocenigo and D. M. Tow, "Managing a Network That Won't Sit Still," *AT&T Bell Laboratories Record* **62**(6), pp. 23-26 (August 1984).

MOOR83. R. L. Moose, Jr., *Link Capacity Assignment in Dynamic Hierarchical Networks for Real-Time Applications,* Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (1983). Masters thesis

MOOR87. R. L. Moose, Jr. and R. E. Nance, "Link Models for Networks with Dynamic Topologies," *Submitted for publication* (1987).

NAGS86. S. Naggapan, *Protocol Design and Analysis for a Dynamic Hierarchical Local Area Network,* Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (1986). Masters thesis

NANR72. R. E. Nance, R. R. Korphage, and U. N. Bhat, "Information Networks: Definitions and Messge Transfer Models," *Journal of ASIS* **23**(4), pp. 237-247 (July-August 1972).

NANR87. R. E. Nance, R. L. Moose, Jr., and R. V. Foutz, "A Statistical Technique for Comparing Heuristics: An Example from Capacity Assignment Strategies is Computer Network Design," *Communications of the ACM* **30**(5), pp. 430-442 (May 1987).

NEUM69. M. F. Neuts, "A Queue Subject to Extraneous Phase Changes," Mimeograph Series No. 207, Dept. of Statistics, Division of Mathematical Sciences, Purdue University (1969). (published in Advances in Applied Probability, 3, 1971, 78-119)

NEUM77. M. F. Neuts, "The M/M/1 Queue with Randomly Varying Arrival and Service Rates," Technical Report No. 77/11, University of Delaware, Dept. of Statistics and Computer Science (1977).

NEUM78. M. F. Neuts, "Further Results on the M/M/1 Queue with Randomly Varying Rates," Technical Report 78/4, University of Delaware, Dept. of Statistics and Computer Science (1978).

NEUM81. M. F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach,* John Hopkins University Press (1981).

POTD76. D. Potier, E. Gelenbe, and J. Lenfant, "Adaptive Allocation of Control Processing Unit Quanta," *Journal of the ACM* **23**(1), pp. 97-102 (January 1976).

PURP74. P. Purdue, "The M/M/1 Queue in a Markovian Environment," *Operations Research* **22**(3), pp. 562-569 (May-June 1974).

PURP78. P. Purdue, "The Single Server Queue in a Random Environment," *Proc. Third Symp. on Operations Research: Methods of Operations Research* **33**, pp. 363-372 (September 1978).

SCOM64. M. Scott, *A Study of Some Single-Counter Queueing Processes,* University of North Carolina at Chapel Hill (1964). Ph.D. dissertation

YANR86. R. M. Yanney and J. P. Hayes, "Distributed Recovery in Fault-Tolerant Multiprocessor Networks," *IEEE Transactions on Computers* **COM-35**(10), pp. 871-879 (October 1986).

YECU71. U. Yechiali and P. Naor, "Queueing Problems with Heterogeneous Arrivals and Service," *Operations Research* **19**(3), pp. 722-734 (May-June 1971).

YECU73. U. Yechiali, "A Continuous-Type Birth-and-Death Process Defined on a Continuous-Time Markov Chain," *Operations Research* **21**(2), pp. 604-609 (1973).

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: SRC-87-011 (TR-87-30 Dept. of C.S.)	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Link Capacity Assignment in Dynamic Hierarchical Networks	5. TYPE OF REPORT & PERIOD COVERED: Interim Report, 1984-86
	6. PERFORMING ORG. REPORT NUMBER: SRC-87-011 (TR-87-30 Dept of C.S.)
7. AUTHOR(s): Richard E. Nance Robert L. Moose, Jr.	8. CONTRACT OR GRANT NUMBER(s): N60921-83-G-A165
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Systems Research Center and Dept. of Computer Science Virginia Tech Blacksburg, VA 24061	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Sea Systems Command SEA61E Washington, D.C. 20362	12. REPORT DATE: July 1987
	13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Naval Surface Weapons Center Dahlgren, VA 22448	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution to Navy agencies and submission for publication in a peer-reviewed journal.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

This report supercedes the original report released May 1985 SRC-85-010 / CS TR-85-25.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer-Communication Networks: Network Architecture and Design -- distributed networks, network topology; Distributed Systems; Local Networks. Design, Performance, Theory. Dynamic topology, link capacity assignment, reconfigurable systems.

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

The dynamic hierarchy, which is a generalization of the centralized, tree structured network, is int oduced in this paper. First, a queueing network model of the dynamic hierarchy is formulated Following the derivation of a network performance measure, probabilistic and heuristic assignment strategies are created. These strategies are compared through the use of analysis of variance procedures and preferred strategies are selected. It is found that a relatively simple heuristic strategy produces capacity assignments whose quality is comparable to that of the preferred probabilistic strategy.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)