A Queueing Network Analysis of Dynamic Reconfigurability in an Hierarchical Information Network*

Richard Nance and U. Narayan Bhat

TR 92-62

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

December 31, 1992

Technical Report SRC-92-010[*]

A QUEUEING NETWORK ANALYSIS OF DYNAMIC RECONFIGURABILITY IN AN HIERARCHICAL INFORMATION NETWORK

U. Narayan Bhat
Department of Statistics
Southern Methodist University
Dallas, Texas 75275

Richard E. Nance[†]
Systems Research Center and
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, Virginia 24061-0251

December 1992

ABSTRACT

Hierarchical information networks are important in applications where the information management must support an existing tree-structured organization. Embedded computer-communication systems in military applications, with a dominant hierarchical command structure, are the most prominent examples. Also typical of such applications is the variability in message demand (both sources and intensity) depending on the external conditions encountered by the encapsulating system (the system supported by the embedded computer-communication system).

Using a queueing network model of an hierarchical information network, we compare the effect of limited dynamic reconfiguration on expected transmission delays. The limited reconfigurability takes the form of apex transition among a proper subset of the communication nodes designated as the apex candidate set. Each apex candidate can assume the ultimate position under designated conditions. This network architecture is termed a *dynamic hierarchy*.

The model includes $N + 1$ nodes $(0, ..., N)$ with 0 identifying the apex node. We assume that message processing at each node is described by a $M/M/1$ model (single server with Poisson arrivals and exponential service times). Further message transfers among the nodes are served by communication links which also behave as $M/M/1$ queues.

Two distinctive features characterize the queueing network model:

(1) The assignment of a set of weights to the nodes dependent on the hierarchical level reflects the increasing importance of information as it is transferred to higher levels.

(2) The dynamic hierarchy requires a communications protocol that partitions the analysis of network delay into three periods: (1) regular operation, (2) reconfiguration, and (3) adjustment. Characterization of the performance of the dynamic hierarchy entails the description of message transmission delay as a composite of the three periods.

1. INTRODUCTION

"Local Area Networks" (LANs) has become an inclusive technical term for describing computer/communication networks requiring high data transfer rates (10-250 Mbps) among a spatially restricted (2 km or less) group of users and processors. From an application perspective, LANs are utilized in office automation, process control, computer assisted engineering and design, and many other areas. Many of these applications emphasize information transfer more than computation, and often in military uses the LAN is embedded within a larger system whose needs are data-intensive. Furthermore, the characteristic parameter values, such as transfer rate and spatial dispersion, are under continual expansion; so that, for the purposes of this paper, we utilize the term "information network" – a computer network having most of the LAN characteristics serving data-intensive needs.

An early classification of information networks recognized some distinct characteristics of hierarchical organization structures (Davis, 1970). The strictly hierarchical information network is described in (Nance, 1972, pp. 245-256), in which the prescriptive nature of message transfer is noted.

During the past ten years the topological design of LANs has been dominated by the bus/tree technology (Stallings, 1985, p. 331). The ring topology has been the only serious alternative, and the linear bus in some circles has been viewed as synonymous with "distributed processing."

Network topologies can take both *logical* and *physical* forms; for example, a logical ring structure could be implemented by a circular connected medium or by a cyclic polling policy used by a network controller. The *communications protocols*, together with the medium capabilities, define the logical topology.

This paper describes a model of a hierarchical information network with limited reconfiguration capability. The motivation for the network architecture and the

protocol specification for dynamic reconfigurability are described in the following section. Section 3 describes the queueing network model of the *dynamic hierarchy*, focusing on mean transmission delay as the primary performance measure. Section 4 and 5 provide the analyses of transmission and nodal delay respectively and Section 6 gives a composite delay measure which can be used for comparison of different forms of dynamic hierarchical structures and for comparing the behaviors of dynamic and static hierarchical structures.

2. THE DYNAMIC HIERARCHY

Originally suggested by Nance (1979) as a hierarchical information and control structure supporting combat systems on Naval surface ships, the dynamic hierarchy concept is introduced in Nance and Moose (1988). The concept varies the conventional tree structured network by allowing any one among a designated set of nodes to assume the apex in response to changing external or internal conditions. The nodes with such capability constitute the Apex Candidate Set (ACS). Figure 1 illustrates the logical topologies that can be produced in a simple 11-node network employing the dynamic hierarchy concept without varying the connectivity relationships. Note that connectivity redefinition is also permitted under the concept, whose utility is described in Derrick and Nance (1986).

The motivating factors for strict hierarchical networks are described in many sources, for example, see Nance, et. al. (1972). In the Naval combat systems domain the topology of the information network mirrors that of the combat hierarchy that it supports. The ability to reconfigure dynamically eliminates a serious weakness in the hierarchical topology: loss of functionality if the apex note is disabled. However, in addition to higher reliability, the concept offers potential advantages in performance.
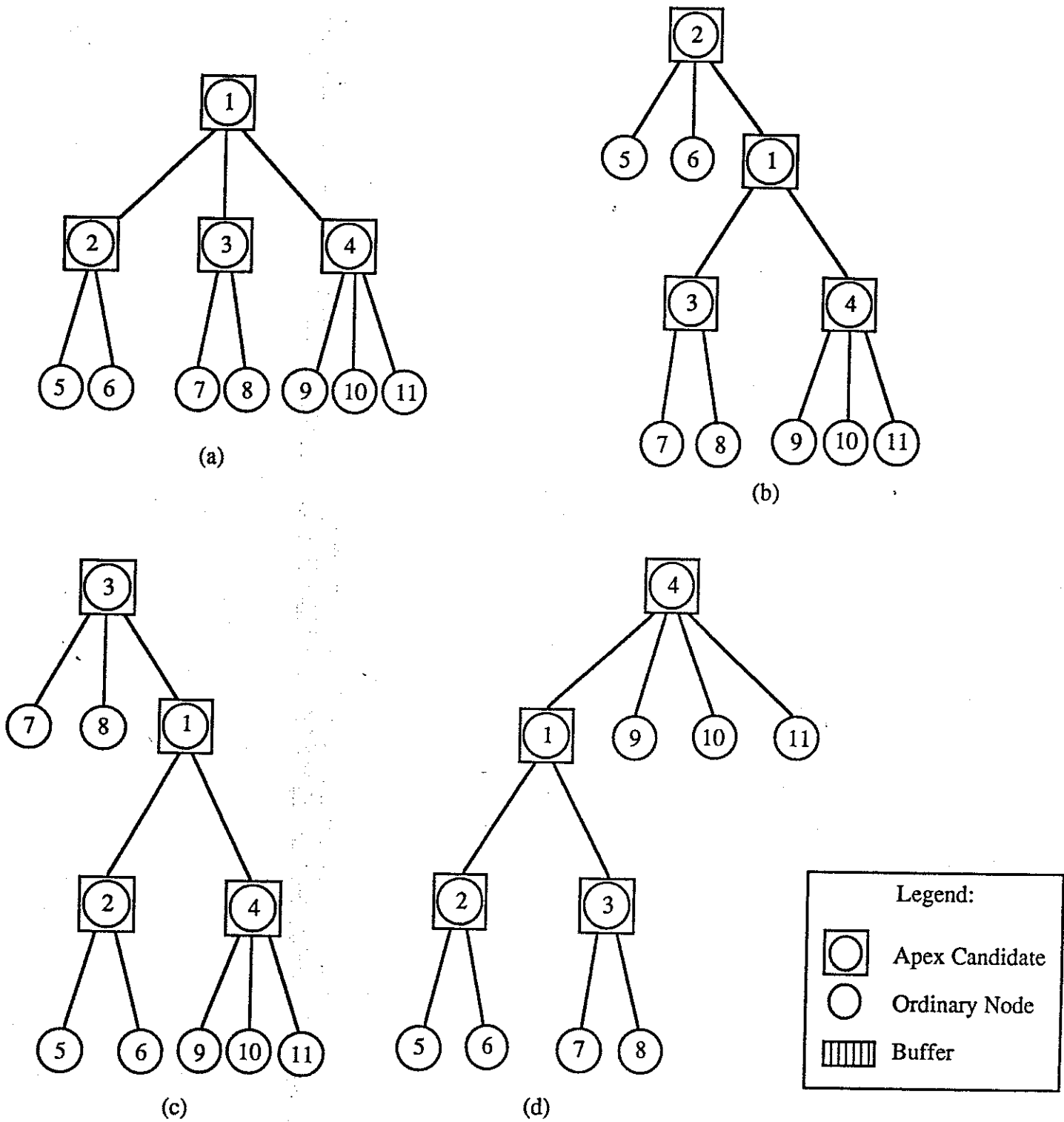
Figure 1. Alternative Topological Configurations in a Dynamic Hierarchical Architecture

Venkateshwaran, et. al. (1988) in a simulation study show the ratio of average message transmission delay for the same static and dynamic networks to increase from approximately 1:1 (6 nodes) to 10:1 (16 nodes).

Adaptive Reconfigurability. Reconfiguration as a recoverability strategy is only one advantage of the dynamic hierarchy. A second advantage lies in the ability to tailor an information network to meet highly variable demands and respond to very different message transfer requirements; for example, see Nance et. al (1987). Without adaptive reconfigurability, certain design decisions would represent a compromise solution. The more variable the requirements, the less satisfactory the compromise in responding to any particular situation.

Adaptive reconfigurability offers the potential for new approaches to adaptability and computer system security. The dynamic hierarchy represents a philosophy of "design for change" that naturally extends to the accommodation of extensibility and adaptability objectives. Relocation of the apex and the replication of data enabling apex reassignment provide opportunities for new innovative protection policies.

Protocol Definition. The model developed in Section 3 requires only a high-level description of the apex transition protocol. A more detailed definition is given in Nagappan (1986).

The hierarchical network of $N+1$ nodes, labeled 0, 1,2,...,N with 0 the apex node, consists of a subset of nodes 0,1,...,k ($k<N$) which form the ACS. All other nodes are linked to one or more nodes in the ACS in a strictly hierarchical topology, i.e. each node, other than the apex, has exactly one link to a node on the immediately higher level. For members of the ACS, a predefined routing table specifies the transmission links to all other ACS members under designated situations. Linkages to the non-ACS members (ordinary nodes) remain fixed.

Message transmission follows the prescribed routing until an event occurs (receipt of a message) signaling the necessity for an apex transition. The transition or

reconfiguration protocol prescribes the following:

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

(2) The incumbent apex node 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* message is sent by each apex candidate to inform the incumbent that a transition is accepted. No acknowledgement represents an unaccepted decision (a "no" vote).

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

(5) Each ACS node, on receipt of the *transition commit* message interrupts data message transfer to all ACS nodes and buffers these messages until the transition period ends. All data message traffic to the ordinary nodes continues, resembling a number of independent hierarchical LANs, which is illustrated in Figure 2.

(6) A *transition over* message is sent by both the former and the new apex signaling the completion of the transition period. Active routing tables are updated, and all previously buffered messages can now be transmitted.

Moose (1989) has obtained approximations of mean network delay under the dynamic hierarchy by considering as network if queues in a random environment that is simplified to enable a matrix-geometric treatment. The more detailed queueing network model described in the following section treats the message delays affected by the above protocol together with the more usual delays associated with link and nodal processing. While ignoring the different classes of messages (control versus data) and the use of packet versus message switching, the model does address the accumulation of "value" or significance as messages ascend to higher levels. The model is intended as a "first cut" in the evaluation of dynamic reconfigurability.

3. THE QUEUEING NETWORK MODEL

The modeling objective is to assess the relative cost/benefit tradeoffs of employing the dynamic versus static hierarchical architecture in embedded information networks.
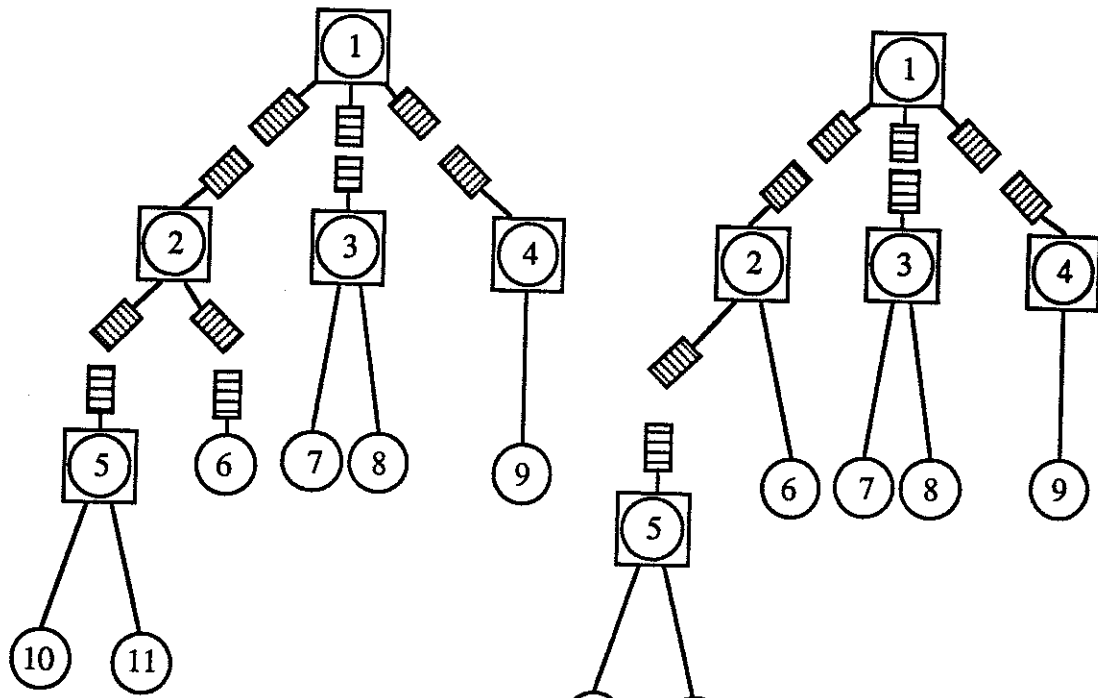
(a) Hierarchy prior to transition
    initiation from Node 1 to Node 2

(b) Hierarchy during transition
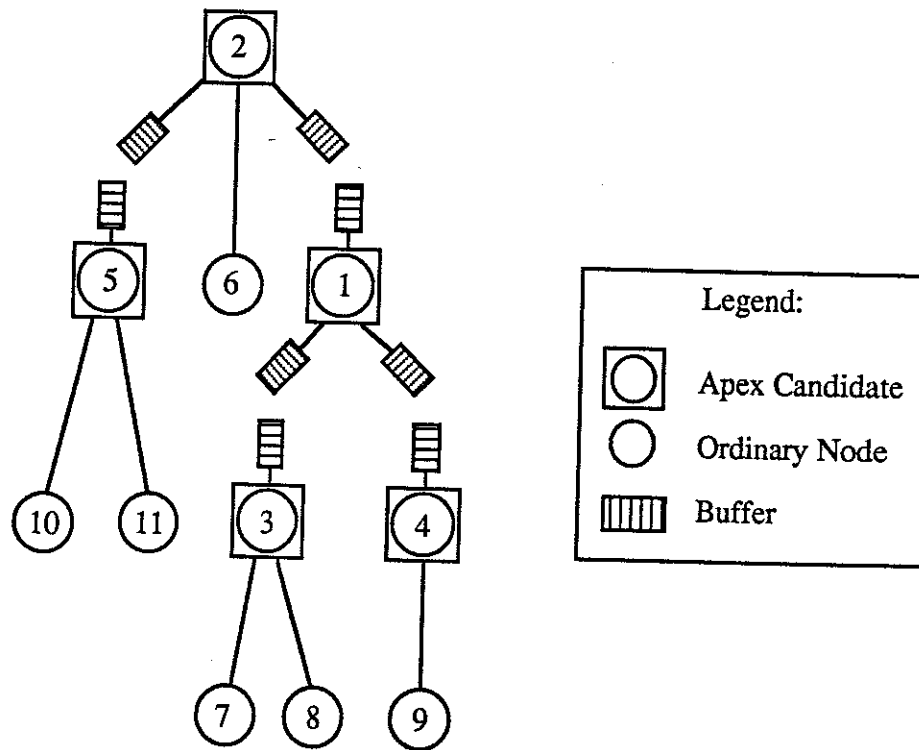
(c) Hierarchy after transition with Node 2 as Apex

Figure 2. Network Topological Structure During Transition
           Illustrating the Message Buffering(from [NAGAS86, p. 22])

As a first step, focus is placed on the average message transmission delay as the behavioral measure of interest. The motivation for dynamic reconfigurability stems from the effect on transmission delays experienced under highly variable demand environments. In the consideration of message transmission delay, the following points should be noted:

(1) Delay encountered at higher levels of the hierarchy carries more weight (is more costly) than delay at lower levels.

(2) Delays are of two types: link delay and nodal delay. Link delay is experienced in the communication between nodes; nodal delay is incurred in the service provided at a node.

(3) In the dynamic hierarchical architecture an apex transition interrupts data message transmission among apex candidate nodes for a time period called the *reconfiguration period*. During this period only local hierarchical transmissions and services are provided, and data messages among ACS members are buffered.

(4) After the reconfiguration period buffered messages among ACS nodes are transferred to their respective destinations on a first come, first served basis. For the ACS nodes the first busy period following reconfiguration consists of the transmission of buffered messages and newly arriving messages. This busy period is called the *adjustment period*. Regular operation is assumed to be in force after the adjustment period.

Characterizing Hierarchical Message Transmission. Comparing the static with the dynamic hierarchy requires the definition of composite delay measures for both architectures. To determine such measures incorporating the delay contributions of all nodes, appropriate weights are identified for each node, initially based on their relative importance. New weights are determined to reflect hierarchical positioning.

Let $s^{(j)}$ be the environment under configuration j and let $w_i^{(j)}$ be the initial weight assigned to node i under $s^{(j)}$. Let the vector of nodal weights be $\underline{w}^{(j)} = (w_0^{(j)}, w_1^{(j)},...,w_N^{(j)})$. To reflect the relative hierarchical position of the node in a path, dynamic weights are determined by accumulating the crude initial weights of nodes comprising the links in the transmission path. When more than one path leads to the same node, the node accumulates weights from all converging paths. This procedure can be formally established as follows.

Consider the environment topology graph C^j and let $H^{(j)}$ be its adjacency matrix with element values 1 at the position (p, q) if a link $p \rightarrow q$ exists and 0 otherwise. Let $H^{(j)2}$, $H^{(j)3},...,H^{(j)n}$ be the successive powers of $H^{(j)}$, where n is the maximum level in the hierarchy. Now the cumulative weights for nodes in $C^{(j)}$, say $\underline{W}^{(j)} = (W_0^{(j)}, W_1^{(j)},..., W_N^{(j)})$, are given by

$$\underline{W}^{(j)} = \sum_{r=0}^{n} \underline{w}^{(j)} H^{(j)r} \quad \text{where } H^{(j)0} \equiv I.$$

This procedure is illustrated below.

Consider a configuration with nodes (0, 1, 2, 3) with initial weights (.8, .6, .4, .2) respectively in parentheses. (We have dropped superscript j denoting environment, for convenience.)

We have $\underline{w} = (.8, .6, .4, .2)$.

$$H = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad H^2 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

and $H^k = 0$ for $k > 2$. Thus we get

$$\underline{W} = \underline{w} H^0 + \underline{w} H + \underline{w} H^2$$

$$= (.8, .6, .4, .2) + (1.0, 0, .2, 0) + (.2, 0, 0, 0)$$

$$= (2.0, .6, .6, .2).$$

Nodal Traffic. Messages arriving at a node can be classified in two different ways.

(1) Based on source: external to the network, or transferred from another node;

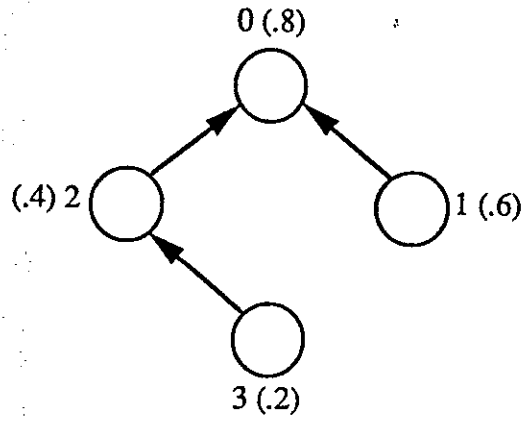(2) based on destination; to be transferred to another node, or to be processed at

Figure 3. Network for Illustrating the Hierarchical Weighting Procedures

the receiving node.

For notational simplicity, the subscript i is used explicitly only when ambiguity might otherwise result. The argument t or d indicates the source of the message, t for transferred and d for external source. Let

γ_i: rate of message traffic to be transferred by node i

λ_i: rate of message traffic to be processed at node i .

Thus $\gamma_i(t)$ represents the rate of transferred message traffic which will again be transferred and $\gamma_i(d)$ represents the rate of external message traffic which will be transferred. We have $\gamma_i = \gamma_i(t) + \gamma_i(d)$. Similarly $\lambda_i(t)$, $\lambda_i(d)$, and λ_i are defined. Let A represent the set of nodes comprising the apex candidate set and $\overline{A}$, the set of non-ACS nodes. Let p_{ki} be the probability that message received at node k has to be transferred to node i, and ξ_i be the fraction of messages transferred to node $i \in A$ from non-ACS nodes $\epsilon \overline{A}$. Then

$$\xi_i = \left(\sum_{k \in \overline{A}} \gamma_k p_{ki} \right) \Big/ \left(\sum_{k} \gamma_k p_{ki} \right)$$

Let

$\beta_i =$ rate of message traffic for processing at node i coming from $\overline{A}$ and external sources

$\quad = \xi_i \lambda_i(t) + \lambda_i(d)$, and

$\alpha_i =$ rate of message traffic to be transferred to node i from ACS nodes.

The messages buffered during a reconfiguration period are assumed to be processed at the beginning of the adjustment period.

Link Traffic. The following notation is used in characterizing message transmission over links.

$a_{ki} =$ traffic rate for transmitted messages on the link from node k to node i during regular operation

$$= \gamma_k p_{ki} \text{ , and}$$

L_A = the set of links joining ACS members.

No messages are transmitted on the link set L_A during reconfiguration. Then a_{ki} is the rate at which the messages are buffered. During adjustment, the buffered messages are transmitted along with the new traffic.

Let ν_{ki} be the transmission rate over the link from k to i and $\rho_{ki} = a_{ki}/\nu_{ki}$. The message transmission time is assumed to be exponential. Also assume T_j as the length of the environment $s^{(j)}$ and T_{rc} as the (mean) length of reconfiguration period.

It is assumed that the delay at every node has simple M/M/1 queue characteristics for the nodal traffic as well as the link traffic. Nodal traffic delay is due to processing required at the node and link traffic delay is due to the time needed for transmission.

As described earlier the network undergoes a reconfiguration (rc) period, an adjustment (ad) period and a period of regular operation (ro) in a cyclical manner. During a reconfiguration period messages are queued for transmission. During the adjustment period messages waiting for transmission from the reconfiguration period as well as any new messages arriving will be processed. This period extends until the end of the busy period. At the end of the busy period regular operation can be assumed to have commenced. We assume T_j, the amount of time environment $s^{(j)}$ continues and T_{rc}, the reconfiguration period within T_j as known. The adjustment period which we shall identify as T_{ad} is essentially the length of the busy period, which we shall denote as BP, initiated by the number of messages arrived during the reconfiguration period. Furthermore simple M/M/1 queueing model is used for both link transmission and nodal processing, and the environment period T_j is assumed to be long enough for stationary results to hold.

Because of the simplicity of this model we concentrate on mean values of transmission and nodal delay, rather than their distribution characteristics. Sections 4

and 5 derive these results for environment $\varepsilon^{(j)}$ (notation for the environment is suppressed for convenience, at this stage) and Section 6 synthesizes global measures across environments to permit a comparative assessment of the dynamic hierarchy.

4. ANALYSIS OF TRANSMISSION DELAY

Let D_{ki} be the transmission delay (service + wait) over the link from k to i. It should be noted that only ACS (Apex Candidate Set) nodes go through a reconfiguration phase in a cycle, while non-ACS nodes operate without any such disturbance. Messages transmitted from ACS nodes face delay due to reconfiguration. Because of these features we shall use the following symbols to represent delay based on when the messages arrive and when they are transmitted.

$E_{ro}[D_{ki}]$ = expected delay during regular operation

$E_Q[D_{ki}]$ = expected delay for messages queued up at the beginning of the reconfiguration period.

$E_{rc}[D_{ki}(B)]$ = expected delay for messages buffered during the reconfiguration period.

$E_{ad}[D_{ki}(N)]$ = expected delay for new messages that arrive during the adjustment period.

We derive expressions for these expected values using results available for M\M\1 and M/G/1 queues.

During regular operation for links in the ACS and at all times for non-ACS links

$$E_{ro}[D_{ki}] = \frac{1}{\nu_{ki}(1 - \rho_{ki})} \cdot \tag{1}$$

During reconfiguration and adjustment periods we must consider the number of messages queued for transmission at their initiations. Assume that Q_{ki} messages are waiting at the start of reconfiguration so that

$$E(Q_{ki}) = \frac{\rho_{ki}}{1 - \rho_{ki}} \cdot \tag{2}$$

Conditional on Q_{ki}, the delay for this group of messages has the following components:

(i) delay before the end of regular operation $\left(\frac{1}{2} \frac{Q_{ki}}{a_{ki}} \right)$.

(ii) reconfiguration period delay (T_{rc}),

(iii) delay during the transition period to allow for a first come first served queue discipline, $\left(= \frac{Q_{ki}(Q_{ki}-1)}{2} \frac{1}{\nu_{ki}} \right)$ and

(iv) channel prorogation (transmission) delay.

Removing the condition, we then get

$$E_Q[D_{ki}] = \frac{E(Q_{ki})}{2 a_{ki}} + T_{rc} + \frac{E(Q_{ki}-1)}{2\nu_{ki}} + \frac{1}{\nu_{ki}} \cdot \tag{3}$$

Substituting from (2) and simplifying we get

$$E_Q[D_{ki}] = \frac{1}{\nu_{ki}(1 - \rho_{ki})} + T_{rc} \cdot \tag{4}$$

Let N_{rc} be the number of messages arriving for transmission during the reconfiguration period. These messages are buffered during the transition for the transmission of Q_{ki} messages, if any. The components of delay for these messages are similar to those queued during reconfiguration considered above. Using similar arguments, we get

$$E_{rc}[D_{ki}(B)] = \frac{T_{rc}}{2} + \frac{E(Q_{ki})}{\nu_{ki}} + \frac{E(N_{rc}-1)}{2\nu_{ki}} + \frac{1}{\nu_{ki}} \cdot \tag{5}$$

Note that $E(N_{rc}) = a_{ki}T_{rc}$.

Following the apex transition is an adjustment period, during which the mean delay of new messages arriving in the first busy period includes the time needed to serve

$Q_B = Q_{ki} + N_{rc}$ messages. Using the arguments of Balachandran (1973) for an M/G/1 queue, this mean delay can be given as

$$E_{ad}[D_{ki}(N)] = \frac{V(S_{ki})}{2E(S_{ki})} + \frac{V(Q_B)E(S_{ki})}{2E(Q_B)} + \frac{E(S_{ki})E(Q_B)}{2} + \frac{a_{ki}E(S_{ki}^2)}{2(1-\rho_{ki})}$$

where S_{ki} is the transmission time over link (k, i) and V is used to denote variance. Writing $E(S_{ki}^2) = 2/\nu_{ki}^2$ and $E(S_{ki}) = 1/\nu_{ki}$, we get

$$E_{ad}[D_{ki}(N)] = \frac{1}{2\nu_{ki}} + \frac{V(Q_B)}{2E(Q_B)\nu_{ki}} + \frac{E(Q_B)}{2\nu_{ki}} + \frac{a_{ki}}{\nu_{ki}(1-\rho_{ki})}$$

$$= \frac{1}{2\nu_{ki}}\left[1 + E(Q_B) + \frac{V(Q_B)}{E(Q_B)} + \frac{2\rho_{ki}}{1-\rho_{ki}} \right], \tag{6}$$

where

$$V(Q_B) = \frac{\rho_{ki}}{(1-\rho_{ki})^2} + a_{ki}\,T_{rc} \quad \text{and} \quad E(Q_B) = \frac{\rho_{ki}}{1-\rho_{ki}} + a_{ki}T_{rc} \quad . \tag{7}$$

Regular operation resumes following the adjustment period. Assuming that the adjustment period ends with the termination of the first busy period, we get

$$T_{ad} = E(BP) = \frac{E(Q_B)}{\nu_{ki}(1-\rho_{ki})}$$

$$= \frac{\rho_{ki}}{\nu_{ki}(1-\rho_{ki})^2} + \frac{\rho_{ki}T_{rc}}{1-\rho_{ki}}$$

$$= \frac{\rho_{ki}}{1-\rho_{ki}}\left[T_{rc} + \frac{1}{\nu_{ki}(1-\rho_{ki})} \right] . \tag{8}$$

Combining expressions from (2), (4), (5), (6) and (7), the overall mean link delay can be expressed as

$$E[D_{ki}] \;=\; \frac{1}{a_{ki}T_j}\Big[E(Q_{ki})E_Q[D_{ki}] \;+\; a_{ki}T_{rc}E_{rc}[D_{ki}(B)] \;+\; a_{ki}T_{ad}E_{ad}[D_{ki}(N)]$$

$$+\;\Big\{(T_j-T_{rc}-T_{ad})a_{ki}-E(Q_{ki})\Big\}\,E_{ro}[D_{ki}]\Big]\;.$$

$$=\;\frac{1}{a_{ki}T_j}\Bigg\{\Big(\frac{\rho_{ki}}{1-\rho_{ki}}\Big)\Big[T_{rc}+\frac{1}{\nu_{ki}(1-\rho_{ki})}\Big]+a_{ki}T_{rc}\Big(\frac{1+\rho_{ki}}{2}\Big)\Big[T_{rc}+\frac{1}{\nu_{ki}(1-\rho_{ki})}\Big]$$

$$+\;a_{ki}T_{ad}\Bigg[\frac{1}{2\nu_{ki}}\Big(1+E(Q_B)+\frac{V(Q_B)}{E(Q_B)}+\frac{2\rho_{ki}}{(1-\rho_{ki})}\Bigg]$$

$$+\;\Big[a_{ki}(T_j-T_{rc}-T_{ad})\;-\;\frac{\rho_{ki}}{1-\rho_{ki}}\Big]\frac{1}{\nu_{ki}(1-\rho_{ki})}\Bigg\}\;, \tag{9}$$

where $E(Q_B)$, $V(Q_B)$ and T_{ad} are given by (7) and (8) above.

Under the assumption of low traffic, we may assume that the contribution to this delay from messages at the start of a reconfiguration period can be ignored. Then the resulting expression is given by

$$E[D_{ki}] \;=\; \frac{1}{a_{ki}T_j}\Bigg\{a_{ki}T_{rc}\Big[(1+\rho_{ki})\frac{T_{rc}}{2}+\frac{1}{2\nu_{ki}}\Big]$$

$$+\;a_{ki}T_{ad}\Big[\frac{1}{\nu_{ki}}+\frac{\rho_{ki}T_{rc}}{2}+\frac{\rho_{ki}}{\nu_{ki}(1-\rho_{ki})}\Big]+\Big[a_{ki}(T_j-T_{rc}-T_{ad})\,\frac{1}{\nu_{ki}(1-\rho_{ki})}\Big]\;,$$

$$\tag{10}$$

where $T_{ad}=\frac{\rho_{ki}T_{rc}}{1-\rho_{ki}}$.

5. ANALYSIS OF NODAL DELAY

Let D be the nodal delay (service + wait) at node i. We assume the service at this node to have an exponential distribution with rate μ. Using expected delay notations similar to those used for transmission delay we derive below expressions for $E_{ro}(D)$, $E_Q(D)$, $E_{rc}[D(N_1)]$, $E_{ad}[D(B)]$ and $E_{ad}[D(N)]$. Additional notations will be introduced as deemed necessary. Except for the analysis necessitated by the special structural differences between transmission and nodal delay, the derivations of these results follow the same arguments as in Section 4. Note that for notational convenience we have suppressed the subscript i in all expressions.

During regular operation for nodes in the ACS and at all times for non-ACS nodes

$$E_{ro}(D) = \frac{1}{\mu(1-\rho)} \quad , \quad \text{where } \rho = \frac{\lambda}{\mu} \quad (\lambda \equiv \lambda_i) . \tag{11}$$

The analysis of reconfiguration delay is more involved.

Assume that a number Q of messages remain to be served at the beginning of the reconfiguration period (using E and V for expectation and variance)

$$E(Q) = \frac{\rho}{1-\rho} \quad \text{and} \quad V(Q) = \frac{\rho}{(1-\rho)^2} . \tag{12}$$

The following two assumptions pertain to data messages within the local hierarchy (the non-ACS nodes served by a single ACS node):

(1) We assume that the reconfiguration period exceeds the busy period initiated with Q local messages to be processed.

(2) We assume that demand on each ACS node after elimination of the backlog of Q messages is reduced to β to reflect the buffering of messages destined for other ACS members until the reconfiguration period ends. Let $\rho_\beta = \beta/\mu$.

Following arguments similar to those used in deriving (3), we determine the average delay for the Q_i waiting messages to be

$$E_Q[D] = \frac{E(Q)}{2\lambda} + \frac{E(Q-1)}{2\mu} + \frac{1}{\mu} .$$
(13)

Substituting values for $E(Q)$ from (12), we get

$$E_Q[D] = \frac{1}{\mu(1-\rho)} .$$
(14)

Mean delay for new customers during the first busy period of the reconfiguration period can be obtained following arguments similar to those for equation (6) ($S \equiv S_i =$ service time at node i).

$$E_{rc}[D(N_1)] = \frac{V(S)}{2E(S)} + \frac{V(Q)E(S)}{2E(Q)} + \frac{E(S)E(Q)}{2} + \frac{\beta E(S^2)}{2(1-\rho_\beta)}$$

$$= \frac{1}{2\mu} + \frac{\rho}{2(1-\rho)^2} \frac{1-\rho}{\rho} \frac{1}{\mu} + \frac{\rho}{(1-\rho)2\mu} + \frac{\beta}{2\mu^2(1-\rho_\beta)}$$

$$= \frac{1}{\mu(1-\rho)} + \frac{\rho_\beta}{\mu(1-\rho_\beta)} .$$
(15)

The mean length of the first busy period is simply

$$\frac{E(Q)}{\mu(1-\rho_\beta)} = \frac{\rho}{\mu(1-\rho)(1-\rho_\beta)} .$$
(16)

Let $E_{rc}[D(N_2)]$ be the mean delay for messages arriving after the first busy period during reconfiguration. Then

$$E_{rc}[D(N_2) = \frac{1}{\mu(1-\rho_\beta)} .$$
(17)

Note that the reconfiguration period extends for a length of time $T_{rc} - \rho[\mu(1-\rho)(1-\rho_\beta)]^{-1}$ after the first busy period.

Note: Under low traffic, if Q can be assumed to be zero, $E_Q[D]$ of (16) can be ignored. Also since ρ_β is likely to be much smaller than ρ, the right hand side of (17) can be approximated by $\frac{1}{\mu}$, which would be the delay for all messages (arriving at rate ρ_β) during reconfiguration. Furthermore, we may assume that no message remains at the end of the reconfiguration period. We use this last assumption in the determination of delay during adjustment. If this assumption is invalid an additional factor equivalent to $\rho_\beta/(1-\rho_\beta)$ must be added to the expected number of buffered messages following the link delay analysis (see Q_B of (7)).

The transition period is the busy period initiated by the messages to ACS members buffered during reconfiguration. Let N_B be the number of buffered messages. Recall that the rate of message traffic transferred to node i from ACS nodes is α_i. We have $E(N_B) = \alpha_i T_{rc}$ and $V(N_B) = \alpha_i T_{rc}$. Following the arguments used in deriving equation (5), mean delay for the buffered messages is given by

$$E_{ad}[D(B)] = \frac{T_{rc}}{2} + \frac{E(N_B-1)}{2\mu} + \frac{1}{\mu}$$

$$= \frac{T_{rc}}{2}(1 + \rho_\alpha) + \frac{1}{2\mu}, \tag{18}$$

where $\alpha/\mu \equiv \alpha_i/\mu_i = \rho_\alpha$.

Following the arguments used in deriving (6), the mean delay for new messages arriving in the first busy period during adjustment is

$$E_{ad}[D(N)] = \frac{V(S)}{2E(S)} + \frac{V(N_B)E(S)}{2E(N_B)} + \frac{E(S)E(N_B)}{2} + \frac{\lambda E(S^2)}{2(1-\rho)}$$

$$= \frac{1}{2\mu} + \frac{1}{2\mu} + \frac{\alpha\,T_{rc}}{2\mu} + \frac{\rho}{\mu(1-\rho)} = \frac{\rho\alpha T_{rc}}{2} + \frac{1}{\mu(1-\rho)} + \frac{1}{\mu}. \quad (19)$$

Also in this case the expected adjustment period is given by

$$T_{ad} = E(BP) = \frac{\alpha\,T_{rc}}{\mu(1-\rho)} = \frac{\rho\alpha\,T_{rc}}{1-\rho}. \quad (20)$$

Note: We have identified two different adjustment periods for links and nodes. If a single adjustment period is to be used appropriate modifications can be made for the expressions.

The environment $\mathscr{E}^{(j)}$ lasts T_j units of time. Allowing for reconfiguration and adjustment, the period of regular operation is $T_j - T_{rc} - T_{ad}$. During this time the mean delay is given by equation (11). Now, combining the various components, the overall nodal delay is given by

$$
\begin{aligned}
E(D) \;=\; & \frac{1}{\lambda T_j}\Bigg\{ E(Q)E_Q[D] + \frac{\rho\rho_\beta}{(1-\rho)(1-\rho_\beta)}\,E_{rc}[D(N_1)] \\[2mm]
& + \beta\left[T_{rc} - \frac{\rho}{\mu(1-\rho)(1-\rho_\beta)} \right] E_{rc}[D(N_2)] \\[2mm]
& + \;\; \alpha\,T_{rc}E_{ad}[D(B)] + \frac{\lambda\rho_\alpha T_{rc}}{1-\rho}\,E_{ad}[D(N)] \\[2mm]
& + \;\left[\lambda(T_j - T_{rc} - T_{ad}) - E(Q) \right] E_{ro}(D) \Bigg\} \\[3mm]
\;=\; & \frac{1}{\lambda T_j}\Bigg\{ \left(\frac{\rho}{1-\rho}\right)\frac{1}{\mu(1-\rho)} + \frac{\rho\rho_\beta}{(1-\rho)(1-\rho_\beta)}\left[\frac{1}{\mu(1-\rho)} + \frac{\rho_\beta}{\mu(1-\rho_\beta)} \right] \\[3mm]
& + \beta\left[T_{rc} - \frac{\rho}{\mu(1-\rho)(1-\rho_\beta)} \right]\frac{1}{\mu(1-\rho_\beta)} + \alpha\,T_{rc}\left[\frac{T_{rc}}{2}(1 + \rho_\alpha) + \frac{1}{2\mu} \right]
\end{aligned}
$$

$$+ \frac{\lambda \rho_\alpha T_{rc}}{1-\rho} \left[\frac{\rho_\alpha T_{rc}}{2} + \frac{1}{\mu(1-\rho)} + \frac{1}{\mu} \right]$$

$$+ \left[\lambda(T_j - T_{rc} - \frac{\alpha\, T_{rc}}{\mu(1-\rho)}) - \frac{\rho}{1-\rho} \right] \frac{1}{\mu(1-\rho)} \Bigg\} . \tag{21}$$

6. COMPOSITE DELAY MEASURES

A distinctive feature of the hierarchical architecture is the relative importance attached to message transmission at different levels. The weighting scheme developed in Section 3 reflects this feature. We have $\underline{W}^{(j)} = (W_0^{(j)},\ W_1^{(j)},\ \ldots,\ W_N^{(j)})$ as the accumulated weights of nodes $(0,\ 1,\ 2,\ \ldots,\ N)$ for the configuration supporting environment $\mathcal{E}^{(j)}$. Link delay is encountered only by messages to be transferred and nodal delay, only by messages to be processed. Consequently, link and nodal contributions to weighted mean delays should be determined individually using $\underline{W}^{(j)}$ and then combined proportionately to give a composite mean delay measure for an environment.

Let $E^{(j)}D_i$ represent mean nodal delay at node i under environment $\mathcal{E}^{(j)}$ and for the $N+1$ nodes let the corresponding mean delays be denoted by

$$\underline{E}^{(j)}D = (E^{(j)}D_0,\ \mathrm{E}^{(j)}D_1,\ldots,\mathrm{E}^{(j)}D_N) .$$

Before combining mean delays with nodal weights, it is necessary to attach weights based on traffic. Let

$$\lambda^{(j)} = \sum_{i=0}^{N} \lambda_i^{(j)}$$

where $\lambda_i^{(j)}$ is the node i traffic under environment $\mathcal{S}^{(j)}$. Define $\eta_i^{(j)} = \lambda_i^{(j)} / \lambda^{(j)}$ and let $H^{(j)}$ be a diagonal matrix with $(\eta_0^{(j)}, \eta_1^{(j)}, ..., \eta_N^{(j)})$ as the diagonal. The composite mean delay $E^{(j)}D$ under environment $\mathcal{S}^{(j)}$ can be given as

$$E^{(j)}D = \underline{W}^{(j)}(\underline{ED}^{(j)}H^{(j)}) . \tag{22}$$

Let π_j $(j = 1,2,...,J)$ be the fraction of time when environment $\mathcal{S}^{(j)}$ is effective. Here we assume that each configuration is used only once. If not, information on the expected number of times of usage is required. The composite mean nodal delay ED_{dyn}^N over all configurations is now obtained as

$$ED_{dyn}^N = \sum_{j=1}^{J} \pi_j E^{(j)} D . \tag{23}$$

Let ED_{dyn}^L be the composite mean link delay over all configurations. It can be determined in an identical fashion by assigning the delay on link (k, i) to node k. Alternatively, we may use the original instead of the accumulated weights.

A Composite Delay Measure. Let $\lambda(L)$ and $\lambda(N)$ be the overall link and node traffic. A combined mean delay measure is obtained as

$$D_{dyn} = \frac{\lambda(L)}{\lambda(L) + \lambda(N)} ED_{dyn}^L + \frac{\lambda(N)}{\lambda(L) + \lambda(N)} ED_{dyn}^N . \tag{24}$$

For the comparable static architecture, the delay expression has the form

$$E(D_i) = \frac{1}{\mu_i(1 - \rho_i)} \tag{25}$$

for node i and

$$E(D_{ki}) = \frac{1}{\nu_{ki}(1 - \rho_{ki})} \tag{26}$$

for link (k, i). Let η_i and η_{ki} be the proportionate traffic at node i and over link (k, i) respectively. Following arguments used in deriving (23), the link and nodal mean delays for a static hierarchy are obtained as

$$ED_{stat}^N = \sum_{i=0}^{N} E(D_i)\eta_i \quad ,$$

$$ED_{stat}^L = \sum_{i=0}^{N} E(D_{ki})\,\eta_{ki}$$

which lead to a single composite measure

$$D_{stat} = \frac{\lambda(L)}{\lambda(L) + \lambda(N)}\ ED_{stat}^L\ +\ \frac{\lambda(N)}{\lambda(L) + \lambda(N)}\ ED_{stat}^N\ . \tag{27}$$

The composite measures given by (24) and (27) permit comparative evaluations.

REFERENCES

Balachandran, K. R. (1973). "Control Policies for a Single Server System," *Management Science*, 19(9), 1013-1018.

Davis, Ruth M. (1970). "The National Biomedical Communications Network as a Developing Structure", *Conference on Interlibrary Communications and Information Networks*, Warrenton, Virginia, Sept. 27-Oct. 2.

Derrick, E. J. and Richard E. Nance (1986). "Local Area Networks and the Dynamic Hierarchy: A Tutorial," Technical Report SRC-86-003, Systems Research Center, Virginia Tech, Blacksburg, Virginia.

Green, P. E. (1979). "An Introduction to Network Architectures and Protocols," *IBM Systems Journal*, 18(2), 202-222.

Moose, Robert L., Jr. (1989). "Modeling Networks with Dynamic Topologies," *ORSA Journal on Computing*, 1(4), 223-231.

Nagappan, Sekkappa (1986). "Protocol Design and Analysis for a Dynamic Hierarchical Local Area Network", M.S. Thesis, Department of Computer Science, Virginia Tech, Blacksburg, VA.

Nance, Richard E. (1979). "Distributed Computing: An Overview", Presentation Series at the Naval Surface Weapons Center.

Nance, Richard E., Robert L. Moose, Jr. and Robert V. Foutz (1987). "A Statistical Technique for Comparing Heuristics: An Example from Capacity Assignment Strategies in Computer Network Design," *Communications of the Association for Computing Machinery*, 30(5), 430-442.

Nance, Richard E. and Robert L. Moose, Jr. (1988). "Link Capacity Assignment in Dynamic Hierarchical Networks," *Computer Networks and ISDN Systems*, 15, 189-202.

Nance, Richard E., Robert R. Korfhage and U. Narayan Bhat (1972). "Information Networks: Definitions and Message Transfer Models," *Journal of the American Society for Information Science*, 23(1), 237-247.

Stallings, William (1985). *Data and Computer Communications*. Macmillan Publishing Company.

Venkateshwaran, Anjali, Richard E. Nance and Osman Balci (1988). "Dynamically Reconfigurable Networks: Concept Evaluation Through Simulation," *Proceedings of the IMACS 12th World Congress on Scientific Computation*, Volume 3, Paris, France, 18-22 July, 460-463.

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
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	Unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S) Systems Research Center SRC-92-010 Blacksburg, Virginia 24061-0251	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) 320 Femoyer Hall, Virginia Tech Blacksburg, Virginia 24061-0251		7b. ADDRESS (City, State, and ZIP Code) Dahlgren, Virginia 22448

8a. NAME OF FUNDING / SPONSORING ORGANIZATION Space and Naval Warfare Systems Command	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code) Washington, DC 20363-5100	10. SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

A Queueing Network Analysis of Dynamic Reconfigurability in an Hierarchical Information Network

12. PERSONAL AUTHOR(S)
U. Narayan Bhat and Richard E. Nance

13a. TYPE OF REPORT Technical	13b. TIME COVERED FROM __n/a__ TO ____	14. DATE OF REPORT (Year, Month, Day) December 1992	15. PAGE COUNT 22

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Design; performance; dynamic reconfiguration; hierarchical information network; dynamic hierarchy; communications protocol

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

Hierarchical information networks are important in applications where the information management must support an existing tree-structured organization. Embedded computer-communication systems in military applications, with a dominant hierarchical command structure, are the most prominent examples. Also typical of such applications is the variability in message demand (both sources and intensity) depending on the external conditions encountered by the encapsulating system (the system supported by the embedded computer-communication system).

Using a queueing network model of an hierarchical information network, we compare the effect of limited dynamic reconfiguration on expected transmission delays. The limited reconfigurability takes the form of apex transition among a proper subset of the communication nodes designated as the apex candidate set. Each apex candidate can assume the ultimate position under designated conditions. This network architecture is termed a *dynamic hierarchy*.

The model includes $N + 1$ nodes $(0, ..., N)$ with 0 identifying the apex node. We assume that message processing at each node is described by a $M/M/1$ model (single server with Poisson arrivals and exponential service times). Further message transfers among the nodes are served by communication links which also behave as $M/M/1$ queues.

Two distinctive features characterize the queueing network model: (1) The assignment of a set of weights to the nodes dependent on the hierarchical level reflects the increasing importance of information as it is transferred to higher levels. (2) The dynamic hierarchy requires a communications protocol that partitions the analysis of network delay into three periods: (1) regular operation, (2) reconfiguration, and (3) adjustment. Characterization of the performance of the dynamic hierarchy entails the description of message transmission delay as a composite of the three periods.

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

DD Form 1473, JUN 86   *Previous editions are obsolete.*   SECURITY CLASSIFICATION OF THIS PAGE