Modeling Networks with Dynamic Topologies

Robert L. Moose Jr.

TR 88-32

Technical Report SRC 88-001

Modeling Networks with Dynamic Topologies†

Robert L. Moose, Jr.

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

January 1988

---

†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.

ABSTRACT

Dynamic hierarchical networks represent an architectural strategy for employing adaptive behavior in applications sensitive to highly variable external demands or uncertain internal conditions. The characteristics of such architectures are described, and the significance of adaptive capability is discussed. The necessity for assessing cost/benefit tradeoffs leads to the use of queueing network models. The general model, a network of $M/M/1$ queues in a random environment, is introduced and then is simplified so that the links may be treated as isolated $M/M/1$ queues in a random environment. This treatment yields a formula for approximate mean network delay by combining matrix-geometric results (mean queue length and mean delay) for the individual links. A discrete event simulation model is defined as a basis for cross-validation of the analytic model. Conditions under which the analytic model is considered valid are identified through comparison of the two models.

CR Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design — *distributed networks, network topology*; C.2.5 [**Computer-Communication Networks**]: Local Networks; G.3 [**Probability and Statistics**]

General Terms: Design, Performance, Theory

Additional Key Words and Phrases: dynamic topology

Modeling Networks with Dynamic Topologies

Robert L. Moose, Jr.

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

1. Introduction

This research addresses the analysis of a class of computer communication networks whose members are referred to as *dynamic hierarchical networks,* or *dynamic hierarchies.* The dynamic hierarchy is an architectural concept that represents a generalization of the conventional tree structured architecture in which the network operates under a centralized, hierarchical mode of control. An overriding characteristic of these conventional (static) hierarchies is that, at the root of a tree-structured topology, there exists a single node that exercises primary control. Secondary capabilities filter down through the remainder of the network in a hierarchical manner. With this basic characteristic in force at all times, if the topology is allowed to vary, the resultant network is a dynamic hierarchical network.

The dynamic hierarchy is intended for applications in which it is possible and desirable to distinguish among multiple *scenarios,* external situations or sets of internal conditions. Each configuration (state of the topology) and apex node of the network corresponds to a different scenario. Configuration changes coincide with changes in the scenario.

The scenario and configuration can be considered dependent on the state of an *environment.* This view leads to a model of the dynamic hierarchy as a network of $M/M/1$ queues in a random network, for each situation.

In some cases, variability of arrival rates is caused entirely or partially by changing conditions inside the network. However, it remains possible, as an abstraction, to view variable internal arrival rates as being caused by variable external arrival rates (and a varying topology).

In either of these cases the arrival rates and the network are considered to be influenced by an external environment. Each state of the environment corresponds to one of the scenarios (set of external arrival rates or internal conditions), which are in one-to-one correspondence with network configurations/potential apex nodes.

The assumptions that define the full model, a network of $M/M/1$ queues in a random environment or *RE-network*, of the dynamic hierarchy include the following:

(1) Scenario changes occur according to a random (Markov) environment process.

(2) The network configuration for each scenario is given.

(3) 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 $\gamma_{jk}^{(i)}$.

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

(5) Nodal processing times are negligible.

(6) Messages pass through the network in a store-and-forward fashion. That is, for each source/destination pair j, k in configuration i, there exists a uniquely specified shortest path $\psi_{jk}^{(i)}$ of links connecting nodes j and k. A message arriving at node j is alternately stored at the source or an intermediate node and transmitted across the next link in $\psi_{jk}^{(i)}$. (The symbol ψ_{jk}, without the superscript (i), is used in later sections to denote the emphasis on logically dynamic hierarchies, in which message paths do not vary with the environment state and configuration). Full receipt of the message is required prior to each forwarding operation. The message departs the network following its delivery to destination node k.

(7) At each node, there exists a separate first-come-first-served queue with unlimited buffer space for each outgoing link.

(8) Propagation times are negligible.

(9) Each node possesses sufficient processing capability to operate all incoming and outgoing links (transmissions) simultaneously.

(10) Links are physically capable of transmission in one direction only. Bidirectional transmission is enabled by connecting nodes with link pairs whose components transmit in opposite directions.

(11) Links are noiseless and error free.

(12) The lengths of arriving messages are exponentially distributed with mean μ^{-1} bits. Additionally, at each intermediate node in the path of a message, upon entering service, the length of the message is reset according to the same exponential distribution (A variant of Kleinrock's independence assumption [KLEL64,KLEL76]).

(13) The message length processes of (12) are mutually independent.

(14) The collections of processes of (3) and (12) are independent of each other.

For routing purposes, messages are assumed to be typed according to their source and destination (nodes). A jk-message, a message with source j and destination k, arrives to node j, follows a fixed (unique for each environment state) path $\psi_{jk}^{(i)}$ of links through the network, and departs after reaching node k.

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, $(\underset{\sim}{N},\ E) = \{\underset{\sim}{N}(t),\ E(t);\ t \geq 0\}$ is a Markov process with state space $(\bigtimes_{l=1}^{L} S_{N_l}) \times S_E$, where $S_{N_l} = \{0,\ 1,\ 2,\ \ldots\}$ is the state space of a single queue length process.

It can be shown that in configuration i, the external arrival process to link l is Poisson with rate $\lambda'^{(i)}_l = \sum_{\{jk:l\epsilon\psi_{jk}^{(i)}\}} \gamma_{jk}^{(i)}$. Briefly, on $\{E(t) = i\}$, the original external arrival processes are independent, Poisson with rates $\gamma_{jk}^{(i)}$. Thus, the superposition of the specified processes is Poisson with rate $\lambda'^{(i)}_l$ [CINE75]. Further, given the state of the environment process, these individual link, external arrival processes are mutually independent and the collection of these processes is independent of the collection of service time (message length) processes. Note that $\lambda_l'^{(i)}$ accounts for arrivals from outside the network only. The composite rate $\lambda_l^{(i)}$ of message arrivals to link l in configuration i includes contributions due to message forwarding inside the network. In general, derivation of the $\lambda_l^{(i)}$'s requires detailed knowledge of the steady state message flow rates.

2.2. Approximate Single Link Models

Nodal processing times in the dynamic hierarchy are assumed negligible and each node is assumed to possess the capability to operate all attached links simultaneously. Thus, links and their queues are the individual service systems of interest. Nodes are considered points at which messages are routed. Two variants of the RE-queue, one with *instantaneous reconfigurations* and one with *reconfiguration periods*, are employed as models of these individual systems.

The following notational conventions are employed in this section and subsequent sections: Symbols in bold type denote vectors. Depending on the context in which they appear, symbols in normal, non-bold type denote matrices or scalars. The symbols **0** and **1** denote the appropriate size vectors of zeros and ones, respectively.

The *link model with instantaneous reconfigurations* is defined as follows. Let $E = \{E(t);\ t \geq 0\}$, the environment process, be a Markov process with state space $S_E = \{1, 2, \ldots, M\}$, irreducible generator Q, and invariant probability vector $\pi = [\pi^{(1)}, \pi^{(2)}, \ldots, \pi^{(M)}]$. On $\{E(t) = i\}$, the composite arrival process to a link j is Poisson with rate $\lambda_j^{(i)}$, the length of a message entering service is reassigned according to an exponential distribution with mean μ^{-1}, message lengths are independent, and the arrival and message length processes are independent. Let C_j be the transmission capacity of link j. Assume that the time to effect a reconfiguration is negligible so that reconfigurations are considered to occur instantaneously with environment state changes.

Then the process $(N_j, E) = \{(N_j(t), E(t));\ t \geq 0\}$, where $N_j(t)$ is the queue length at time t, is an RE-queue with arrival rate vector $\boldsymbol{\lambda}_j = [\lambda_j^{(1)}, \lambda_j^{(2)}, \ldots, \lambda_j^{(M)}]$ and service rate vector $\boldsymbol{\mu}_j = [\mu C_j, \mu C_j, \ldots, \mu C_j]$. This joint process has a block-tridiagonal generator $\tilde{Q}_j$. For the stable queue $\left(\frac{1}{\mu C_j}\pi\boldsymbol{\lambda}_j < 1\right)$, the stationary queue length/environment probabilities are given by [NEUM81, p.258]

$$\boldsymbol{x}_k = \pi(I - R_j)R_j^k, \quad k \geq 0,$$

where R_j is the minimal solution of

$$R_j^2 \Delta(\boldsymbol{\mu}_j) + R_j[Q - \Delta(\boldsymbol{\lambda}_j + \boldsymbol{\mu}_j)] + \Delta(\boldsymbol{\lambda}_j) = 0$$

The vector $\boldsymbol{x}_k$ has elements $x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(M)}$, where $x_k^{(i)}$ is the stationary probability that the queue length is k and the environment state is i.

Mean virtual waiting time in the queue is [NEUM81, pp. 133-134]

$$W_{Vj} = \pi(I - R_j)\sum_{k=1}^{\infty} R_j^k \sum_{\nu=0}^{k-1}[-(Q - \Delta(\boldsymbol{\mu}_j))^{-1}\Delta(\boldsymbol{\mu})]^{\nu}[-(Q - \Delta(\boldsymbol{\mu}_j))]^{-1}$$

$$\cdot\, [-(Q - \Delta(\boldsymbol{\mu}_j))^{-1}\Delta(\boldsymbol{\mu}_j)]^{k-\nu}\mathbf{1}$$

Mean virtual time at link j (time in the queue plus service time) is

$$S_{Vj} = W_{Vj} + (\pi\boldsymbol{\mu}_j)^{-1}$$

For a dynamic hierarchy in which reconfigurations cannot be considered instantaneous, the *link model with reconfiguration periods* is defined. In such a network, each reconfiguration of the network topology is preceded by a reconfiguration period, during which the network performs the processing necessary to enable operation in the new configuration. Transmission of messages on links connecting potential apex nodes ceases (messages are buffered) during a reconfiguration period. Operation of links not connecting potential apex nodes and operation of all links at times not in reconfiguration periods proceeds as previously described.

This behavior is modeled by first adding the *reconfiguration states* $1_0, 2_0, \ldots, M_0$ to the state space of the environment process to form the state space S_{E_1}. The generator of the environment process E_1 is defined so that a transition into a state $i \in \{1, 2, \ldots, M\}$ is preceded by a sojourn in state i_0. Thus, a sequence of environment transitions has the form

$$\cdots \; i_0 \rightarrow i \rightarrow j_0 \rightarrow j \rightarrow k_0 \rightarrow k \; \cdots .$$

Configuration i_0, which corresponds to environment state i_0, is identical to configuration i. The external arrival rates $\{\gamma_{jk}^{(i_o)}\}$ in configuration i_0 are equal to those in configuration i. However, for each link j that connects potential apex nodes, the service rate on j in configuration i_0 is zero. The service rate on j in configuration i is μC_j.

This link model is an RE-queue with server interruptions. If link j connects potential apex nodes, then $\boldsymbol{\lambda}_j = [\lambda_j^{(1_o)}, \lambda_j^{(1)}, \lambda_j^{(2_o)}, \ldots, \lambda_j^{(M_o)}, \lambda_j^{(M)}]$ and $\boldsymbol{\mu}_j = [0, \mu C_j, 0, \mu C_j, \ldots, 0, \mu C_j]$. Otherwise, $\boldsymbol{\lambda}_j = [\lambda_j^{(1_o)}, \lambda_j^{(1)}, \lambda_j^{(2_o)}, \ldots, \lambda_j^{(M_o)}, \lambda_j^{(M)}]$ and $\boldsymbol{\mu}_j = [\mu C_j, \mu C_j, \mu C_j, \mu C_j, \ldots, \mu C_j, \mu C_j]$. The previously discussed results regarding the stationary queue length/environment probabilities and mean virtual time at a link hold with the modified environment process generator, arrival rate vector, and service rate vector. The stability condition is $\dfrac{\boldsymbol{\pi}\boldsymbol{\lambda}_j}{\boldsymbol{\pi}\boldsymbol{\mu}_j} < 1$, where $\boldsymbol{\pi} = [\pi^{(1_o)}, \pi^{(1)}, \pi^{(2_o)}, \pi^{(2)}, \ldots, \pi^{(M_o)}, \pi^{(M)}]$ is the invariant probability vector of this environment process.

2.3. Network Model with the Departure Process Assumption

Exact analysis of the RE-network model of Section 2.1 through the multidimensional Markov process $(\underline{N}, E)$ is beyond the scope of this research. Further, the network is not separable in the sense that the product form networks of Kelly [KELF79] (or others) are separable. To decompose the network into a collection of single server (RE-) queues, an additional assumption is introduced. Results for single queues are obtained by viewing them as if they were RE-queues in isolation. The single queue results are combined to produce a characterization of network performance. Note that for the remainder of the analysis it is assumed that reconfigurations are instantaneous. That is, the first link model of Section 2.2 is employed. However, the network model is valid and the analysis is similar when the link model with reconfiguration periods is used.

The key assumption leading to a decomposition of the network is:

Departure process assumption: Assume that at each queue, the process of jk-message departures is statistically identical to the process of jk-message arrivals to that queue.

The following result now holds.

Theorem 1.

Given the departure process assumption

(1) For each link jk, each flow process of jk-messages in the network is a Poisson process in a random environment (an *RE-Poisson* process).

(2) The jk-arrival processes to a link l, for all jk such that $l \in \psi_{jk}$, are mutually independent given the environment state.

(3) The composite arrival process to each link is an RE-Poisson process.

proof:

First, the following characteristic of logically dynamic hierarchies is reiterated and its implication is noted: A jk-message passes through the network over a unique, shortest path ψ_{jk}. Since ψ_{jk} is the shortest path from node j to node k, it contains no repeated links. Once a message traverses ψ_{jk} and reaches node k, it departs the network. Thus, indirect (and direct) feedback is not possible. Further, since the network is tree-structured and all queueing is FCFS with single servers, overtaking is not possible. Complications due to feedback and overtaking therefore are absent from these networks.

The validity of (1) is established by observing that as a result of the departure process assumption, the jk-arrival process at node j, which is a Poisson process in a random environment by definition, propagates through the network to become a collection of RE-Poisson processes of jk-messages arrivals to and departures from each link in the path ψ_{jk}. The members of this collection are statistically identical to each other and to the jk-arrival process at node j.

Result (2) follows from the departure process assumption and the tree-structured, non-overtaking nature of the network. By definition, the external jk-arrival processes, for all jk, are mutually independent given the environment state. For a fixed jk, nowhere in the network does the flow of jk-messages experience cycles or overtaking. Further, for any $l \in \psi_{jk}$, there exists only one jk-arrival process to queue l. This jk-arrival process is the propagation to queue l of the external jk-arrival process to node j. The external jk-arrival processes are mutually independent given the environment state. It follows from the departure process assumption that the propagations of the members of any subset of this collection of processes are mutually independent given the environment state. Specifically, for each l, this independence property holds among the jk-arrival processes to queue l, for jk such that $l \in \psi_{jk}$.

The third result follows from (1) and (2). For any l, for each jk such that $l \in \psi_{jk}$, on $\{E(t) = i\}$ the jk-arrival process to queue l is a Poisson process with rate $\gamma_{jk}^{(i)}$ and the elements of the collection of such processes at queue l are mutually independent. Thus, on $\{E(t) = i\}$ the composite arrival process to link l is a Poisson process with rate $\lambda_l^{(i)} = \sum_{jk:l \in \psi_{jk}} \gamma_{jk}^{(i)}$. Therefore, the composite arrival process to link l is an RE-Poisson process with rate vector $\lambda_l = [\lambda_l^{(1)}, \lambda_l^{(2)}, \ldots, \lambda_l^{(M)}]$.

■

Corollary 1

Given the departure process assumption, queue l in an RE-network is an RE-queue with arrival rate vector $\boldsymbol{\lambda}_l = [\lambda_l^{(1)}, \lambda_l^{(2)}, \ldots, \lambda_l^{(M)}]$, where $\lambda_l^{(i)} = \sum_{jk: l \in \psi_{jk}} \gamma_{jk}^{(i)}$.

2.4. Mean Network Delay

Theorem 1 and Corollary 1 show that the departure process assumption has the desired effect of simplifying a network (in a random environment) of queues of unknown probabilistic nature to a network of RE-queues with known arrival rates (The service rates are known in any case.) In this section, the individual RE-queue results, which are summarized in Section 2.2, are combined to derive a formula for approximate mean network delay.

For each $j \in \{1, 2, \ldots, L\}$, let $\boldsymbol{\lambda}_j = [\lambda_j^{(1)}, \lambda_j^{(2)}, \ldots, \lambda_j^{(M)}]$ and $\boldsymbol{\mu}_j = [\mu C_j, \mu C_j, \ldots, \mu C_j]$ be the arrival rate and service rate vectors, respectively. Assume that $\boldsymbol{\pi}\boldsymbol{\lambda}_j < \boldsymbol{\pi}\boldsymbol{\mu}_j$ and let R_j be the rate matrix for queue j, that is, the minimal solution of

$$R_j^2 \Delta(\boldsymbol{\mu}_j) + R_j[Q - \Delta(\boldsymbol{\lambda}_j + \boldsymbol{\mu}_j)] + \Delta(\boldsymbol{\lambda}_j) = 0.$$

Then the stationary environment/queue length probabilities for queue j are

$$x_{jk} = \pi(I - R_j)R_j^k, \;\; k \geq 0$$

and mean virtual time at queue j is

$$\bar{S}_{Vj} = \pi(I - R_j)\sum_{k=1}^{\infty} R_j^k \sum_{\nu=0}^{k-1} [-(Q - \Delta(\boldsymbol{\mu}_j))^{-1}\Delta(\boldsymbol{\mu}_j)]^{\nu}[-(Q - \Delta(\boldsymbol{\mu}_j))]^{-1}$$
$$\cdot\, [-(Q - \Delta(\boldsymbol{\mu}_j))^{-1}\Delta(\boldsymbol{\mu}_j)]^{k-\nu}\mathbf{1} + (\boldsymbol{\pi}\boldsymbol{\mu}_j)^{-1}.$$

Define the following additional notation:

$W_{jk} =$ network delay of a jk-message.

$T_l =$ delay of a message at queue l.

$T =$ network delay of a message.

$p_{jk} =$ probability that a message is a jk-message.

Then mean network delay is

$$\bar{T} = E[T]$$

$$= \sum_j \sum_k E[T \mid jk\text{—message}] P(jk\text{—message})$$

$$= \sum_j \sum_k p_{jk} E[W_{jk}], \qquad \text{on } \{jk\text{—message}\} \;\; T = W_{jk}$$

$$= \sum_j \sum_k p_{jk} \sum_{l \in \psi_{jk}} E[T_l], \qquad \text{since } W_{jk} = \sum_{l \in \psi_{jk}} T_l.$$

The mean arrival rate of jk-messages is $\sum_{i=1}^{M} \pi^{(i)} \gamma_{jk}^{(i)}$ and the mean arrival rate of all messages is $\sum_j \sum_k \sum_{i=1}^{M} \pi^{(i)} \gamma_{jk}^{(i)} = \sum_{i=1}^{M} \pi^{(i)} \sum_j \sum_k \gamma_{jk}^{(i)} = \sum_{i=1}^{M} \pi^{(i)} \gamma^{(i)}$, where $\gamma^{(i)} = \sum_j \sum_k \gamma_{jk}^{(i)}$.

Now let

$$p'_{jk} = \mathit{weight} \text{ of } jk\text{-traffic}$$

$$= \frac{\sum_{i=1}^{M} \pi^{(i)} \gamma_{jk}^{(i)}}{\sum_{i=1}^{M} \pi^{(i)} \gamma^{(i)}}$$

and suppose that mean time at queue l, $E[T_l]$, can be approximated by mean virtual time at queue l, $\bar{S}_{Vl}$. Then mean network delay is approximated by

$$T' = \sum_j \sum_k p'_{jk} \sum_{l \in \psi_{jk}} \bar{S}_{Vl}$$

$$= \sum_j \sum_k \left(\frac{\sum_{i=1}^{M} \pi^{(i)} \gamma_{jk}^{(i)}}{\sum_{i=1}^{M} \pi^{(i)} \gamma^{(i)}} \right) \sum_{l \in \psi_{jk}} \bar{S}_{Vl}$$

$$= \sum_{l=1}^{L} \Big(\sum_{i=1}^{M} \pi^{(i)} \gamma^{(i)} \Big)^{-1} \bar{S}_{Vl} \sum_j \sum_{\substack{k \\ jk:\, l \in \psi_{jk}}} \sum_{i=1}^{M} \pi^{(i)} \gamma_{jk}^{(i)}$$

$$= \sum_{l=1}^{L} \frac{\bar{\lambda}_l}{\sum^{M} i = 1\, \pi^{(i)} \gamma^{(i)}} \bar{S}_{Vl},$$

where

$$\bar{\lambda}_l = \sum_j \sum_{\substack{k \\ jk:\, l \in \psi_{jk}}} \sum_{i=1}^{M} \pi^{(i)} \gamma_{jk}^{(i)}$$

$$= \sum_{i=1}^{M} \pi^{(i)} \sum_{j} \sum_{\substack{k \\ jk:\, l \in \psi_{jk}}} \gamma_{jk}^{(i)}$$

$$= \sum_{i=1}^{M} \pi^{(i)} \lambda_l^{(i)}.$$

Note that $\lambda_l^{(i)}$ is the composite arrival rate to queue l when the environment state is i. Thus, $\bar{\lambda}_l$ is the mean composite arrival rate to queue l.

3. Simulation Model

A simulation model is employed in this investigation to provide confidence interval and point estimates that are used to assess the accuracy of the mean network delay approximation. Specifically, in this cross-validation effort, the results of the two models are compared to determine the conditions under which the approximation is accurate.

Many details may be included in a simulation model. Decisions to include or exclude details should be guided by the study objectives. The objectives of this study are stated roughly in the previous paragraph. The accuracy of the mean network delay approximation is affected by the departure process assumption and the use of mean virtual time at a queue to approximate mean actual time at a queue. For the comparison, a source is needed of results that may be viewed as those of an *exact* RE-network (without the departure process assumption or the use of virtual time at a queue). Thus, the dynamic hierarchy simulation model is intended to represent the behavior of an RE-network. It is important to note that the simulation results remain model results. Neither the simulation nor the analytic model produce exact results for the dynamic hierarchy, the real system of interest.

3.1. Model Structure

In line with the study objectives, the structure of the simulation model closely reflects that of the analytic model. The components (objects or processes) of the model are *message generator, message, node, link,* and *environment.* Nodes are passive and hold routing tables that identify the next node and link in the path of a message. Nodes are connected (conceptually) by links, which, as in the analytic model, are the servers. Each link contains an FCFS message queue. A message enters the network at its source node, waits (possibly) and receives service at the links in its path, and departs the network at its destination node. The environment generates a Markov process of environment states that affect the network as described with respect to the analytic model. Message generators supply messages according to RE-Poisson processes.

3.2. Model Implementation

The dynamic hierarchy simulation model is implemented in the SIMSCRIPT II.5 † simulation programming language using the process interaction world-view [RUSE83]. This implementation and the simulation experiments are done on the IBM system at Virginia Tech using the CMS and MVS operating systems. The program consists of approximately 1025 lines of code (and in-line documentation) plus approximately 170 lines of introductory documentation.

Implementation counterparts of the previously mentioned model components are defined as *PROCESS*(es) and *ENTITY*(s) in this program. Additionally, various attributes and routines are defined at the *SYSTEM* level. *ROUTINE*(s) include those for implementation of the environment transition mechanism, input/output, and support of the experiment design.

3.3. Experiment Design

The method of batch means is employed for experimentation and output analysis. Each run consists of a transient period and multiple, consecutive batches.

Estimation of the transient period length is accomplished through simulation runs with a simple test network network and various sets of parameters. Six pilot runs are executed. Each has a simulated run length of 1200 seconds. Runs 1, 2, and 3 correspond to one set of parameters

---

† The original implementation was written in Simula 67 [BIRG73] but subsequently was translated to SIMSCRIPT II.5.

and runs 4, 5, and 6 correspond to another. Within each group, {1,2,3} and {4,5,6}, different runs are defined by using different stream values (or equivalently, using different initial seeds) for the SIMSCRIPT random variate routines.

Each pilot run is divided into batches of 250 messages. Within a batch, an observation is accumulated for mean queue length at each queue, mean delay at each queue and mean network delay. At the end of each batch a moving average is calculated (updated) for each of these means. In this way, the run yields a collection of sequences of time/moving average pairs— one sequence for each of the means of interest.

The results of the pilot runs are compiled in graphs that are used to estimate the length of the transient period in general. Two such graphs are shown in Figure 1. Figure 1(a) plots the moving average for mean network delay in runs 1, 2, and 3. Figure 1(b) plots the moving average for mean network delay in runs 4, 5, and 6. Differences among the curves in a single graph are due to statistical variation among multiple runs of the simulation with different random number streams. The time to reach steady state, that is, the transient period length is visually estimated. (In Figure 1, this time is indicated by a vertical line. The individual transient period length estimates are compiled in Table 1. The average value in this table is 683.2071 seconds and the maximum is 938.6364 seconds.

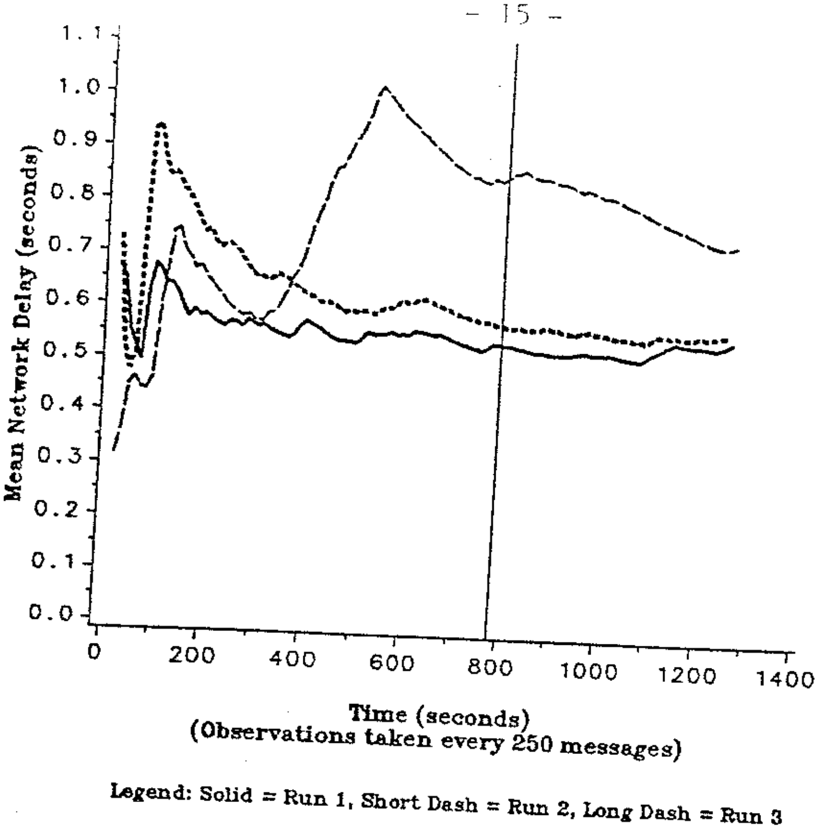

Figure 1(a)
Moving Average of Mean Network Delay: Pilot Runs 1 to 3

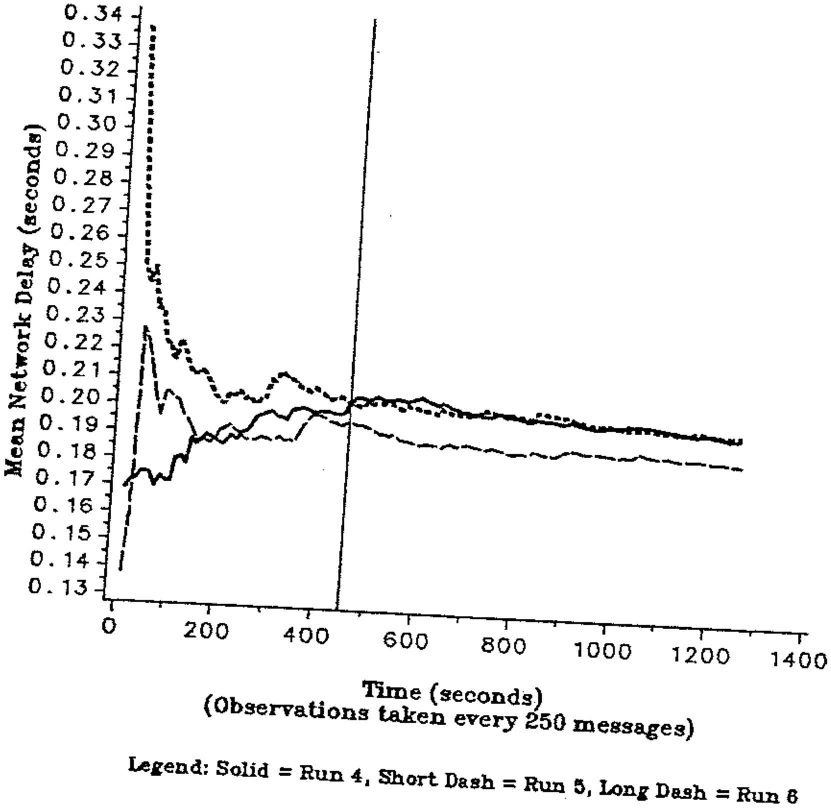

FIGURE 1(b)
Moving Average of Mean Network Delay: Pilot Runs 4 to 6

Figure 1. Moving Average Graphs For Determination of Transient Period Length

Table 1. Observed Times to Reach Steady State

performance measure	transient period length
network delay	784.8485
queue length	456.0606
	774.2424
	636.3636
	784.8485
	710.6061
	572.7273
	938.6364
	434.8485
queue delay	615.1515
	790.1515
	636.3636
	742.4242
	715.9091
	562.1212
	848.4848
	721.2121
	572.7273

As a conservative estimate that is expected to be sufficiently large for all cases of interest, 1200 seconds is chosen as the transient period length. For this network, approximately 24300 to 29400 messages pass through the network in that time. Therefore, in subsequent simulation runs, it is assumed that steady state conditions hold upon the departure of the 30000th message. Observations accumulated during the transient period are deleted.

The value of 30000 messages is chosen as the batch size as well. Within each batch, an observation is accumulated for the mean and variance of queue length and delay at each link and the mean and variance of network delay. After the final batch, these observations are used to calculate point and confidence interval estimates for mean queue length at each link, mean delay at each link, and mean network delay.

3.4. Dependence of the Batch Means

It is believed that the statistical observations derived from a simulation generally are not independent (See, for example [LAWA82, p. 279]). Thus, ". . . classical statistical analyses based on IID observations are not directly applicable" [LAWA82, p. 279] in such situations and may yield inaccurate results. With a batch means experiment design, the correlation among these observations and the resultant adverse effects may be reduced by selecting a large batch size. The authors expect the batch size of 30000 messages employed in the dynamic hierarchy simulation to reduce the correlation among the batch means sufficiently so that valid conclusions may be drawn.

3.5. Credibility Assessment

The steps taken to establish the credibility [BALO87] of the simulation model include: (1) design of the model in a structured manner, (2) implementation in a well known simulation programming language, (3) extensive examination (desk checking) of the implemented code, (4) execution tracing, and (5) stress testing.

Regarding steps (1) and (2), the model was designed in a top-down fashion and subsequently implemented in SIMSCRIPT II.5. The resultant model structure and implementation are described briefly in Sections 3.1 and 3.2, respectively, and in greater detail in [MOOR88]. At various points in the implementation, debugging, and initial experimentation, the code was examined (entirely or in part) to insure its agreement with the model design.

To confirm that the execution of the implemented model behaves as expected, a trace is generated of selected events in a simulation of a simple test network. The events traced are: environment activation and transition, message activation, message arrival to a node, message arrival to a link, and message departure from the network. source or destination.

Examination of the trace output shows, for example, that messages follow the correct path, all messages eventually depart the network, no simultaneous arrivals occur, no message departs from a link at the same time it arrives at the link (that is, no service time is zero), no two messages depart from the same link simultaneously, and messages do not overtake one another.

Last, the environment events in the trace show that all intertransition times of the environment are nonzero, and no one step transitions of the form $i \rightarrow i$ occur.

For stress testing, the network topology, environment parameters, capacities, and mean message length are identical to those used for runs 4, 5, and 6 in the determination of the transient period length. Different traffic matrices are generated such that utilization of the most heavily utilized link (link 4) takes on the values 0.9, 0.95, 0.99, 0.995, and 0.999 in successive experiments. The remaining links experience a corresponding increase in utilization.

The simulation output from these experiments does not show any unexpected behavior. As expected, as the network load increases and ρ_4 approaches 1, mean queue length and mean link delay increase rapidly and would become unmanageably large for ρ_4 sufficiently close to 1 (theoretically approaching ∞ as $\rho_4 \rightarrow 1$). These mean values and the value of mean network delay are given in Table 2. Although mean network delay experiences a significant increase, it remains relatively small due to the moderating influence of the other links, whose utilization factors are not close to 1.

Table 2. Network Performance Under Stress Testing

link 4			network
utilization	mean queue length	mean delay	mean delay
0.90	9.9069	1.0966	0.4963
0.95	19.0543	2.0029	0.7707
0.99	165.8173	16.5834	4.7146
0.995	221.5103	22.1765	6.1974
0.999	301.3271	30.1893	8.2966

4. Comparison of Analytic and Simulation Models

As previously discussed, the purpose of the simulation model is to provide a means of validating the analytic model. More precisely, the comparison enables identification of dynamic hierarchy topologies and parameters for which the analytic model yields accurate results by determining those cases in which the analytic results fall within a 95-percent confidence interval generated by the simulation.

4.1. Test Networks

Topologies for the ten dynamic hierarchies are employed as a basis for the comparison. The networks range in size from three nodes (and four links) to fourteen nodes (and twenty-six links). Some of these networks are used to test certain hypotheses and others are intended to represent typical dynamic hierarchies. For example, networks 1, 6, and 7 are used to examine behavior in nonbranching networks in which messages arrive only to the end nodes and are destined only to the end nodes. Networks 8, 9, and 10 are intended to represent typical dynamic hierarchies. That is, they possess multilevel tree structures and are of moderate size.

Definition of the test networks is completed by specifying various sets of parameters for each network. A set of parameters for a network consists of values for traffic matrices, environment process generator (and the resultant stationary probability vector of the environment process), link capacities, and mean message length. An *experiment* consists of applying the simulation model and analytic model to one topology and parameter set (The terms *topology* and *network* will be used interchangeably where no ambiguity will result.)

The parameter sets are used to examine network behavior under conditions of interest and to represent those of typical dynamic hierarchies. All of the parameter sets are used to test the effect of different levels of link utilization. Approximate values of link utilization are listed in Table 3 [†]. Each network has at least one experiment with medium (0.4 - 0.5) link utilization and at least one with high (0.85 - 0.9) link utilization.

[†] Due to the method of selecting arrival rates and capacities, link utilization factors within a given experiment tend to be approximately equal.

Table 3. Representative Link Utilization Levels

network	parameter set	link utilization
1 to 7	1	0.4444
	2	0.2222
	3	0.8889
8	1	0.4444
	2	0.2222
	3	0.8889
	4	0.3555
	5	0.1807
	6	0.7230
9	1	0.4444
	2	0.4444
	3	0.8889
	4	0.8889
10	1	0.4444
	2	0.4444
	3	0.8889
	4	0.8889

In the parameter sets for networks 1 to 7, the arrival rate matrices are specified so that messages arrive at end nodes and depart from end nodes. For example, in network 3, messages destined for node 1 arrive at nodes 3, 4, and 5, and messages destined for nodes 3, 4, and 5 arrive at node 1. Neither external arrivals nor departures to outside the network occur at node 2. In other words, the network handles 3,1-, 4,1-, 5,1-, 1,3-, 1,4-, and 1,5-traffic only. In networks 1, 6, and 7, this type of traffic pattern enables examination of behavior as a single message stream (in each direction) is propagated down a series of links. In networks 1 to 5, this type of pattern enables examination of the effect of combining separate message streams into a new stream that is transmitted across a single link and the effect of splitting a single message stream into different streams that are transmitted across separate links.

An additional aspect examined in the experiments on network 9 and network 10 is the effect of increasing rates of environment transitions. For each of these networks, experiments 1 and 3 are performed with a specified environment process generator (which is different for each network). Then, experiments 2 and 4 are performed on each network with an environment process generator that is equal to five times the original generator. In this way, the stationary probability vector of the environment process remains the same as the mean frequency of environment transitions increases.

4.2. Comparative Results

The experimentation and comparison proceed as follows. For each experiment a simulation is run. This run yields point and 95-percent confidence interval estimates for mean queue length at each link, mean delay at each link, and mean network delay. Then the analytic model is used to derive numerical values for mean queue length and mean delay at each link and mean network delay for this experiment. A numerical value from the analytic model is considered *accurate* (with respect to the simulation results) if it is contained in the corresponding simulation generated confidence interval. Otherwise, it is considered *inaccurate* and the percentage difference between it and the simulation generated point estimate is computed.

Tabulated results for mean network delay for all test networks are given in Table 4. The columns of this table contain the following data: network number, parameter set number, analytic value for mean network delay (*matrix-geometric network T*), simulation point estimate of mean network delay (*simulation network T*), endpoints of simulation confidence interval for mean network delay (*confidence interval left/right*), indication of whether the analytic value lies within the confidence interval (*T in conf. interval*), and for inaccurate analytic values, the percentage difference between the analytic value and the simulation point estimate relative to the simulation point estimate (*% matrix-geometric T from simulation*).

Table 4. Comparison of Analytic and Simulation Models (Mean Network Delay)

network	parameter set	matrix geometric network T	simulation network T	confidence interval left	confidence interval right	T in conf. interval	% matrix geometric T from simulation
1	1	0.1827	0.1840	0.1830	0.1851	no	0.7065
1	2	0.1266	0.1232	0.1205	0.1259	no	2.7957
1	3	1.0766	1.0402	0.9967	1.0837	yes	-
2	1	0.1205	0.1229	0.1223	0.1235	no	1.9528
2	2	0.0842	0.0847	0.0843	0.0851	no	0.5903
2	3	0.6934	0.6804	0.6406	0.7201	yes	-
3	1	0.1168	0.1194	0.1184	0.1203	no	2.1776
3	2	0.0817	0.0826	0.0823	0.0829	no	1.0896
3	3	0.6486	0.6516	0.6228	0.6805	yes	-
4	1	0.0794	0.0817	0.0810	0.0823	no	2.8152
4	2	0.0556	0.0562	0.0559	0.0565	no	1.0676
4	3	0.4548	0.4630	0.4432	0.4829	yes	-
5	1	0.0999	0.1072	0.1063	0.1081	no	6.8097
5	2	0.0683	0.0699	0.0695	0.0702	no	2.2890
5	3	0.6492	0.6768	0.6450	0.7086	yes	-
6	1	0.1917	0.1904	0.1893	0.1915	no	0.6828
6	2	0.1342	0.1334	0.1330	0.1339	no	0.5997
6	3	1.0647	0.9978	0.9641	1.0314	no	6.7048
7	1	0.3343	0.3403	0.3376	0.3429	no	1.7632
7	2	0.2328	0.2342	0.2315	0.2369	yes	-
7	3	1.7200	1.5463	1.4793	1.6133	no	11.2333
8	1	0.0268	0.0270	0.0269	0.0272	no	0.7407
8	2	0.0190	0.0191	0.0190	0.0192	yes	-
8	3	0.1455	0.1492	0.1398	0.1587	yes	-
8	4	0.0237	0.0246	0.0244	0.0247	no	3.6595
8	5	0.0181	0.0185	0.0184	0.0185	no	2.1622
8	6	0.0701	0.0711	0.0690	0.0733	yes	-
9	1	0.0279	0.0284	0.0282	0.0286	no	1.7606
9	2	0.0276	0.0278	0.0276	0.0280	yes	-
9	3	0.1702	0.1685	0.1614	0.1755	yes	-
9	4	0.1444	0.1452	0.1370	0.1534	yes	-
10	1	0.0120	0.0120	0.0120	0.0121	yes	-
10	2	0.0119	0.0119	0.0118	0.0119	yes	-
10	3	0.0655	0.0639	0.0612	0.0666	yes	-
10	4	0.0609	0.0615	0.0596	0.0635	yes	-

Additionally, similarly structured tables (not included here) of results for the individual links are constructed. Each table compares the matrix-geometric results for mean queue length and mean delay for each link with the corresponding confidence interval and point estimates from the simulation model.

Various conclusions can be drawn from these comparisons. The primary positive conclusions are:

(1) The analytic results are accurate for larger and more complex networks due to the effects of mixing (superposing) internal messages flow processes with external message arrival processes.

(2) The analytic results are accurate under higher levels of link utilization

$(0.85 < \rho_j < 1.0)$.

(3) The analytic results may become more accurate as the frequency of environment transitions increase.

(4) In cases where the analytic results are marginally inaccurate or marginally accurate, the departure process assumption leads to accurate analytic results for mean queue length but the additional error induced by using mean virtual time at a link to approximate mean (actual) time at a link causes inaccuracy in the analytic results for mean link delay.

The primary negative conclusions are (See also the comment in (4) regarding mean link delay):

(5) The analytic results are inaccurate for simple networks, such as networks 1 to 7. For tandem networks, such as networks 1, 6, and 7, the inaccuracy becomes more pronounced as the length (number of links from one end to the other) increases.

(6) The analytic values are inaccurate under low and medium levels levels of link utilization $(0.0 < \rho_j < 0.75)$.

Conclusion (5) is supported by the comparisons for networks 1 to 5, parameter sets 1 and 2, and all but one of the experiments on networks 6 and 7. Except in network 7, parameter set 2 †, analytic mean network delay is inaccurate in all cases. The analytic mean queue length and mean link delay results are mixed. In the experiments on networks 2 to 5, analytic mean queue length generally is accurate. In the experiments on networks 1, 6, and 7, this mean generally is inaccurate. Analytic mean link delay is inaccurate in these experiments.

These inaccuracies are attributed to the absence of mixing of internal flow processes and external arrival processes and to the accumulation of error along message paths in networks 1 to 7. For any jk and path ψ_{jk}, the only node in this path at which external arrivals occur is j. Thus, the flow of jk-messages through the network is never mixed with any external arrival processes. That is, the composite arrival process to each link is either a superposition of multiple flow processes of this type, or one component of a decomposition of a flow process of this type.

It appears that as such processes are propagated through the network without being

---

† Network 7, parameter set 2 is considered an anomalous case and does not affect the observations and conclusions.

influenced by subsequently encountered external arrival processes, the composite departure and arrival processes at each link behave less like the RE-Poisson processes addressed in Theorem 1. The departure process assumption does not reflect the behavior of the network with sufficient accuracy in these types of networks.

The most severe inaccuracies occur in the experiments on the tandem networks 1, 6, and 7. In these networks, a single external arrival process (in each direction) passes through the network, behaving increasingly less like an RE-Poisson process at successive links in its path, especially as the path length increases. Thus, the RE-queue model is less valid, which leads to inaccurate results.

Comparisons in one or more experiments for each network except network 10 support conclusion (6). The utilization levels in these experiments are low to medium. Although the analytic mean queue length and mean link delay results are mixed, analytic mean network delay is inaccurate, and overall, the results for these experiments are considered inaccurate. This conclusion is applicable primarily to relatively simple networks but holds also in more complex networks at low utilization levels.

Although conclusions (5) and (6) are characterized as *negative,* they are useful and significant in the sense that they identify situations in which the analytic model does not apply. Further, by elimination and in combination with the positive conclusions, conclusions such as (5) and (6) assist in identifying situations in which the analytic model does apply.

Conclusions (1) and (2) are the positive counterparts of (5) and (6), respectively. Regarding the effect of the size and complexity of the network, the comparisons indicate a trend toward increasingly accurate results as larger and more complex networks are considered. Networks 8, 9, and 10 represent three steps in increasingly complex, multilevel network topologies. Analytic mean queue length and mean link delay are accurate overall and analytic mean network delay is accurate in three of six experiments (one at low and two at high utilization) on network 8. The analytic results are accurate in all but one of the experiments on networks 9 and 10 (link utilization in these experiments is medium and high).

It is concluded that the higher level of mixing (superposing) of internal flow processes and external arrival process in the more complex networks is responsible for this trend toward increased accuracy. In networks of this type, the flow of *jk*-messages traverses multiple links and passes through multiple nodes. At each intermediate node in this path of *jk*-messages, the *jk*-message flow is mixed with various other internal flow processes and external arrival processes to form the composite arrival process to a specific link. It appears that this mixing, especially with external arrival processes, decreases the distance between the composite arrival process and the assumed RE-Poisson process. This influence offsets the accumulation of error that is observed in the simple networks and thus contributes to more accurate analytic results.

The effect of link utilization level on accuracy of the analytic results is seen in experiments on all of the networks except networks 6 and 7. In all cases (except networks 6 and 7), the analytic results are accurate at a high level of utilization. However, the comparisons do not indicate a trend of increased accuracy in the experiments at medium utilization over the experiments at low utilization.

This behavior is similar to that observed with other heavy traffic approximations in queueing systems (see, for example, the survey discussion in [DISR75]), although the theoretical justification is absent here. It appears that as the probability of reaching the boundary (queue length equal to zero) becomes smaller (as utilization increases), the effects of boundary behavior diminish and the analytic and simulation results become less sensitive to the given distributional assumptions. The consequence is that the two sets of results are in closer agreement, which leads to the conclusion that the analytic values are more accurate.

Conclusion (3) is not supported as well (as the other conclusions) by the comparisons. However, it appears to hold in the cases considered. In two of the networks (9 and 10), parameter sets 1 and 3 use a specified environment process generator. Parameter sets 2 and 4 use a generator that is equal to five times the original generator, thus increasing the transition rates of the environment. The comparisons for these experiments show slight increases in accuracy of the analytic results under higher environment transition rates.

A possible explanation for these increases is that, as in the case of heavy traffic, the analytic and simulation results become less sensitive. Under higher environment transition rates, mean link delay, mean queue length, and mean network delay in the two models approach common values. Thus, the analytic values are considered more accurate.

Lastly, conclusion (4) addresses accuracy in the analytic results for the individual links. As reflected by the number of values that lie within the respective confidence intervals, analytic mean queue length tends to be more accurate than analytic mean link delay. This characteristic is most noticeable in experiments for which overall accuracy (or inaccuracy) is marginal. In these marginal cases, the departure process assumption leads to accurate results for mean queue length. However, the additional error introduced by approximating mean actual time at a link with mean virtual time at a link causes the results for mean link delay to be inaccurate.

5. Conclusions

In assessing the relative costs and benefits of the dynamic hierarchy, a new concept in reconfigurable network architectures, an analytic, queueing model and a discrete event simulation model are developed. The simulation model is used in a cross-validation effort to establish the conditions under which the analytic model is valid. The analytic model is used to derive estimates of performance, as measured by mean network delay, in the dynamic hierarchy, thus eliminating the need to run a potentially costly simulation for each network of interest.

The dynamic hierarchy is modeled as a network of $M/M/1$ queues in a random (Markov) environment, or RE-network. Introduction of an additional assumption, the departure process assumption, enables the links in the network to be analyzed separately as $M/M/1$ queues in a random environment, or RE-queues.

Two variants of the RE-queue are defined as applicable models of the individual links. The first, which the analysis described herein employs, is referred to as the link model with instantaneous reconfigurations and is appropriate both when the time to perform a network configuration change is negligible and as a performance baseline for the second variant. The second is referred to as the link model with reconfiguration periods. It represents a more realistic model in cases where a network configuration change takes a nonnegligible amount of time.

The individual links are analyzed by applying results from the literature on RE-queues for queue length probabilities and mean virtual waiting time. Mean virtual time at a link (time in the queue plus service time) is used to approximate mean actual time at a link. These results for the individual links then are combined to produce the desired formula for mean network delay.

Using selected of dynamic hierarchy topologies with various sets of network parameters, results are obtained from the analytic and simulation models. Then the two sets of results are compared. An analytic value is considered accurate if it lies within the corresponding, simulation generated confidence interval. On the basis of this comparison, the following conclusions are drawn:

(1) The analytic results are accurate for larger and more complex networks and inaccurate for smaller, less complex networks.

(2) The analytic results are accurate under high levels of link utilization $(0.85 < \rho_j < 1.0)$ and inaccurate under low and medium levels of link utilization $(0.0 < \rho_j < 0.75)$.

(3) The analytic results may become more accurate as the frequency of environment transitions increase.

(4) In cases where the analytic results are marginally inaccurate or marginally accurate, the analytic results for mean queue length tend to be less accurate than the analytic results for mean link delay.

Further, the positive effect of higher network complexity offsets the negative effect of a low utilization level and the positive effect of a high utilization level offsets the negative effect of low network complexity in many cases.

These conclusions lead to the following characterization of the conditions under which the analytic results (particularly mean network delay) are expected to be accurate:

> The analytic results are accurate for multilevel networks of at least medium complexity and under high levels of link utilization. Accuracy is maintained for simple networks under sufficiently high link utilization levels and for sufficiently complex networks under low and medium link utilization levels.

Dynamic hierarchies with point-to-point, logically variable topologies are the networks of primary interest in the research discussed herein. The results and conclusions, in addition, can be viewed as characterizing a performance baseline for dynamic hierarchies of other topological classes. In particular, dynamic hierarchies with topologies based on a common transmission medium (for example, a bus), in which (logical) link capacities may be considered variable, are expected to exhibit performance increases over the dynamic hierarchies considered here.

REFERENCES

BALO87. Balci, O., "Credibility Assessment of Simulation Results: The State of the Art," In *Proc. of the Conf. on Methodology and Validation (1987 Eastern Simulation Conf's.)*, Orlando, Florida, pp. 19-25, The Society for Computer Simulation (April 6-9, 1987).

BIRG73. Bitrwistle, G. M., O.-J. Dahl, B. Myhrhaug, and K. Nygaard, *Simula Begin*, New York, Petrocelli/Charter (1973).

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

DISR75. Disney, R. L., "Random Flow in Queueing Networks: A Review and Critique," *AIIE Trans.* **7**(3), pp. 268-288 (September 1975).

KELF79. Kelly, F. P., *Reversibility in Stochastic Networks*, New York, John Wiley & Sons (1979).

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

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

LAWA82. Law, A. M. and W. D. Kelton, *Simulation Modeling and Analysis*, New York, McGraw-Hill (1982).

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

MOOR88. Moose, R. L. Jr., "Analysis of Networks with Dynamic Topologies," Doctoral dissertation, Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, Virginia (1988).

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

RUSE83. Russell, E. C., *Building Simulation Models with SIMSCRIPT II.5*, Los Angeles, CACI (1983).

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Limited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
Systems Research Center SRC-88-001
Blacksburg, Virginia 24061

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Systems Research Center

6b. OFFICE SYMBOL *(If applicable)*

7a. NAME OF MONITORING ORGANIZATION
Naval Surface Warfare Center

6c. ADDRESS *(City, State, and ZIP Code)*
Virginia Tech
320 Femoyer Hall
Blacksburg, Virginia 24061

7b. ADDRESS *(City, State, and ZIP Code)*
Dahlgren, Virginia 22448

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Naval Surface Warfare Center

8b. OFFICE SYMBOL *(If applicable)*

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
N60921-83-G-A165-

8c. ADDRESS *(City, State, and ZIP Code)*
Dahlgren, Virginia 22448

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
Modeling Networks with Dynamic Topologies

12. PERSONAL AUTHOR(S)
Robert L. Moose, Jr.

13a. TYPE OF REPORT
Interim Research

13b. TIME COVERED
FROM ________ TO ________

14. DATE OF REPORT *(Year, Month, Day)*
January 1988

15. PAGE COUNT
30

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

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

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

Dynamic hierarchical networks represent an architectural strategy for employing adaptive behavior in applications sensitive to highly variable external demands or uncertain internal conditions. The characteristics of such architectures are described, and the significance of adaptive capability is discussed. The necessity for assessing cost/benefit tradeoffs leads to the use of queueing network models. The general model, a network of M/M/1 queues in a random environment, is introduced and then is simplified so that the links may be treated as isolated M/M/1 queues in a random environment. This treatment yields a formula for approximate mean network delay by combining matrix-geometric results (mean queue length and mean delay) for the individual links. A discrete event simulation model is defined as a basis for cross-validation of the analytic model. Conditions under which the analytic model is considered valid are identified through comparison of the two models.

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