Purdue University

Purdue e-Pubs

Department of Computer Science Technical Reports

Department of Computer Science

1989

Approximate Delay Analysis of Transaction Oriented Communication in a Distributed System

Dan C. Marinescu

Report Number:

89-875

Marinescu, Dan C., "Approximate Delay Analysis of Transaction Oriented Communication in a Distributed System" (1989). *Department of Computer Science Technical Reports.* Paper 744.
https://docs.lib.purdue.edu/cstech/744

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries.
Please contact epubs@purdue.edu for additional information.

AN APPROXIMATE DELAY ANALYSIS OF TRANSACTION ORIENTED COMMUNICATION IN A DISTRIBUTED SYSTEM

Dan C. Marinescu

CSD-TR-875
March 1989
(Revised November 1990)

To appear in the International Journal of Computer Simulation 1991

AN APPROXIMATE DELAY ANALYSIS OF TRANSACTION ORIENTED COMMUNICATION IN A DISTRIBUTED SYSTEM

*Dan C. Marinescu**

Computer Sciences Department
Purdue University
West Lafayette, IN 47907, USA

Abstract

A methodology for computing the response time in heterogeneous distributed systems is presented. The approximate analysis is based upon two conjectures: (a) the queueing delays in data transmission in a transaction oriented system can be approximated by the queueing delays in an equivalent network with independent data arrival rates at all nodes, and (b) a server entity can be modeled either as an M|M|1 system or as an M|G|1 system depending upon the distribution of the remote service time. These approximations are validated through simulations for a transaction oriented system build around a token ring network.

1. INTRODUCTION

The distributed systems considered in this paper consist of a set of hosts interconnected by a high speed Local Area Network, LAN. In such systems a large fraction of the traffic is duc to *transaction oriented communication.* The transaction oriented communication, also called *request-response communication,* is used in conjecture with the client-server paradigm to move the data and to distribute the computations in the system by requesting services from remote servers [3]. The typical sequence of events in requesting a service from a remote server is: a client entity, a process, task or thread of control, sends a request to a server entity on a remote host, then a computation is performed by the server entity, and, finally a response is sent back to the client.

The quality of service in a transaction oriented system is characterized by the *response time* defined as the time elapsed between the instant a request is generated and the instant when the response comes back to the client entity. The response time is the sum of the queueing delay, the transmission time and the propagation delay of the request, the waiting time and the service time at the remote server, and finally the queueing delay, the transmission time and the propagation delay of the response. A distributed system must satisfy response time requirements. Clearly, the response time

* Research supported in part by the Strategic Defense Initiation under ARO grants DAAG03-86-K-0106, DAAL03-90-0107, by NATO under grant 891107, by the NSF Software Engineering Research Center (SERC), and by NSF grant NCR - 8702115

depends upon the communication load of the network and the computation load of the remote server. To determine if a distributed system satisfies such requirements it is necessary to relate the quality of service measured by the delay with the quantity of service measured by the throughput.

The transaction oriented communication is used to support resource sharing in a distributed system and to implement high level network services. For example, diskless workstations use shared file servers for paging and for file access, X-window terminals connect to shared X-window servers, network file systems provide simultaneous access to a collection of file systems physically connected to different hosts. These and numerous other applications are based upon the *Remote Procedure Call, RPC,* protocols that implement request-response communication. The transaction oriented communication is used by a variety of applications. The request rates, the distribution of the message size, and of the remote execution time differ from one type of application to another. For example, in case of X-windows applications, requests are generated rather frequently, for every X-window "event", relatively short packets are transmitted, and fine grain remote computations are performed most of the time. Paging over the network typically occurs at lower rates, involves larger packets determined by the page size, as well as coarse grain remote execution. The response time in a transaction oriented system is determined by functional requirements. For example, when an RPC protocol is used by a diskless workstation for paging, it is expected that a 'page fault' will take a time comparable to the case when paging is done to a local disk.

There are numerous studies of request-response protocols [1], [4], and their application to high level network services [3], [5], but there are only a few results concerning the analysis of request-response communication. The delay analysis of request-response communication poses several difficult problems. First of all, a salient feature of all models of computer networks is *the independence assumption* that stipulates that the arrival process at each node of the network is independent upon the arrival process at all other nodes. In case of request-response communication, there is a clear, causal relationship, namely, a response is generated only as a result of processing a request, hence it seems unrealistic to extend the independence assumption to this case. Without this assumption, it is difficult to estimate the communication delays required to compute the response time. The second difficult problem is to estimate the remote execution time. Generations of requests can still be modeled as a Poisson process, but request arrival at a remote server is a more general process because a random time elapses between the instant a request is generated and the instant it arrives at the remote server. This random time is affected by the *Medium Access Control* method used for communication in the LAN. If the arrival process at a remote sever can be approximated as a Poisson process then the analysis of the remote execution is much simpler. It follows that an exact delay analysis for the systems with request-response communication is an extremely difficult task, and well justified approximations are necessary for practical design of distributed systems using request-response communication.

The contribution of this paper is a methodology for computing the response time in a distributed system. The approximate delay analysis proposed is based upon two independence assumptions:

A1 *The Communication Independence Assumption.* The queueing delays in the data transmission in a network with request-response communication can be approximated as the queueing delays in an equivalent network with independent data arrival rates at all nodes. The two networks are equivalent, in the sense that they use the same Medium Access Control method and the traffic generated at any node of one network is identical to the traffic generated at the corresponding node of the second network.

A2 *The Remote Execution Independence Assumption.* A server entity can be modeled as an M|M|1 system in case of an exponential distribution of the remote service time or as an M|G|1 system in the general case.

Approximations like the ones presented above are unlikely to be valid for all system configurations and any load, but for particular system configurations subject to load within some range. A simulation experiment to study the effects of the approximations (A1) and (A2) for different system configurations and load is presented in detail in Section 5. Two categories of systems, communication-bound and balanced systems and identified based upon the ratio of the average channel and server utilization. The load is characterized by the request arrival rate and by the traffic load. Several classes of systems, called Type A, B and C are studied separately under different traffic conditions, for different client to server node ratios and using different service disciplines, exhaustive and 1-limited. The results indicate that the Communication Independence assumption is quite satisfactory for the system built around a token ring for the light to medium traffic, and that the Remote Execution Independence assumption is quite robust for the exhaustive service discipline, when the server utilization does not exceed 75%.

The intuitive justification of these assumptions is based upon the following arguments. A server entity acts as a global system resource for a large client population and the arrival process at the server entity results from merging a number of independent Poisson stream arrivals, assumption (A2). The queueing delay of any request arriving at a server entity, depends upon a large number of uncorrelated random variables related to the other requests. This makes the time when a response is generated at a server entity less dependent upon the time when the request was generated, assumption (A1).

A similar independence assumption has been suggested for modeling communication links in a store and forward network [7]. Kleinrock has conjectured that merging several packet streams on a transmission line has an effect akin to restoring the independence of interarrival times and packet lengths and an M|M|1 queueing model can be used for a link. Kleinrock's independence approximation is reasonably good for moderate to heavy traffic loads in a densely connected network for Poisson stream arrivals and exponentially distributed packet length.

The paper is organized as follows. A transaction oriented system built around a token ring is discussed in Section 2 and the model of the system is introduced in Section 3. Section 4 covers the approximate delay analysis of the system, and a simulation experiment designed to validate the approximations proposed in this paper are presented in Section 5.

2. REMOTE PROCEDURE CALL PROTOCOLS AND TRANSACTION ORIENTED COMMUNICATION IN A HIGH SPEED TOKEN RING

The Remote Procedure Call protocols are widely used for transaction oriented communication in a distributed environment. Such protocols are typically implemented at the transport or at a higher layer of a computer communication architecture. VMTP, a transport level protocol specifically designed to support transaction oriented communication was proposed in [4].

Existing RPC protocols are inefficient and difficult to analyze. Mechanisms like flow control and error control appear repeatedly at the Data Link, the Network, and the Transport layers. The queuing delays, the packet transmission time and the propagations delays are generally much smaller than the processing delays for existing protocols. The processing delays account for most of the communication latency, for example, the measurements performed on a rudimentary RPC protocol using UDP as the transport mechanisms in Berkeley Unix 4.2 show that about 11,000 instructions are executed at each end for a *null remote procedure call* with virtually no remote computation and very short messages transmitted [8]. It is also very difficult to analyze high level protocols because they lead to queueing models consisting of networks of queues, [10].

The transaction oriented communication has several important properties that can be used to build simpler and more efficient, *light-weight* protocols. First of all, the request-response is a semantic oriented communication. The client and the server entities exchange messages with a semantic significance to them. A client is interested in receiving a semantically correct response. The error control mechanisms exploiting this property of the request-response communication lead to simpler protocols. The transaction oriented communication has a built-in flow control, the client blocks waiting for a response, and this can be used to further simplify the RPC protocols. The transaction oriented communication could benefit from the addressing schemes that support efficiently high level services in a distributed system. For example to achieve mobility of server entities, location independent addressing is desirable. A *functional addressing scheme* based upon a functional communication model was proposed in [9]. In this scheme a request for service is addressed to "whoever may perform function x, subject to restriction y", rather than to "node z". The functional addressing can be used for achieving dynamic load balance. Such *symmetric systems* (see Section 3.1.b) can be analyzed more easily. The transaction oriented communication is well suited for high speed networks, which allow considerably larger *data transmission units, DTUs*. The high speed of the communication channel allows exchange of possibly large objects without the need and the overhead of breaking them into pieces.

The models of request-response communication presented below consider that the response is the only form of acknowledgment used for error and flow control. In this case the response time depends upon the Medium Access Control method used, and a separate analysis must be performed for different types of LANs. The case of a token ring is examined in this paper as a first example. The token ring is has several interesting properties which make it attractive for distributed systems.

(a) A token ring accommodates well the asymmetry built into a system with request-response communication. In such a system two classes of nodes with very different communication requirements can be identified. The server nodes, acting as global system resources are expected to exhibit heavy incoming and outgoing traffic. The client nodes are expected to have lighter traffic. A token ring with exhaustive service at server nodes, is capable of absorbing this asymmetry.

(b) The delay-throughput characteristics of a token ring match the requirements of a distributed system using request-response communication. Indeed, the response time of a Remote Procedure Call is generally expected to be close to the one, when the remote procedures are executed locally. On the other hand, the traffic in the network is expected to be fairly heavy due to the frequent use of various network services and to the resource sharing. The token ring provides the lowest delay at a given load for the heavy traffic region.

(c) The token ring technology is well established and widely available. The high speed rings, with speed of 100 Mbps and above, are available today.

To reflect the asymmetry of the system different service disciplines, exhaustive and 1-limited service are used at different nodes.

3. THE MODEL OF THE SYSTEM

The token ring network and the cyclic server models have been investigated in the past and important results are reported in [2], [6], [7], [11], [12] and [13]. The analytical model presented below uses the pseudo-conservation laws formulated in [2] because they allow a mix of service disciplines not possible to analyze with other methods. When only the exhaustive service disciplines are used at all nodes of a network, very good approximations of the queuing delays in communication are formulated elsewhere, for example in [6].

In this section the model of the system and the relevant modeling assumptions are described. The communication aspects are examined first and two systems, an asymmetric system and a symmetric one are discussed. Then modeling assumptions relevant to the remote execution supported by the request-response communication are presented.

3.1 The modeling assumptions - the communication side

The relevant assumptions concerning the configuration of the system as well as the communication aspects are summarized in the following.

1. The system consists of $M + N$ client nodes called C-nodes and N server nodes called S-nodes. This model corresponds to a system with M nodes that host only the client entities, and N nodes hosting both client and server entities. An example of such a system is a set of M workstations (client nodes) connected with N servers. Each server can, in turn, be a client to another server, hence a server is modeled as a pair consisting of a C-node and an S-node.
2. A time slotted channel is assumed. Each slot size is equal to the transmission time of a DTU.
3. The arrival processes at each queue is assumed to be independent of the arrival process at other queues. The arrival process at all queues are considered to be

Bernoulli processes with batch arrivals. The first and the second moments of the arrival rates are assumed to be known.

4. The traffic generated at each C-node is due to requests. At each C-node, C_i, $1 \le i \le N + M$, there are N queues $Q_{i,j}$, $1 \le j \le N$ one for each server. The first moment of the arrival process at each queue $Q_{i,j}$ is $\lambda_{i,j}^{req}$. The first moment of the total request rate at C_i is

$$\lambda_i^{req} = \sum_{j=1}^{N} \lambda_{i,j}^{req} \tag{3.1}$$

Each request may consist of one or more DTUs. Denote

$$\begin{aligned} b_{i,j}^{req} &= \text{the number of DTUs per request for } Q_{i,j} \\ \beta_{i,j}^{req} &= E[b_{i,j}^{req}] \end{aligned} \tag{3.2}$$

Note that β_i^{req} is defined as

$$\beta_i^{req} = \frac{\sum_{j=1}^{N} \lambda_{i,j}^{req} \, \beta_{i,j}^{req}}{\lambda_i^{req}} \tag{3.3}$$

The mean number of DTUs per request is

$$\beta^{req} = \frac{\sum_{i=1}^{M+N} \lambda_i^{req} \, \beta_i^{req}}{\sum_{i=1}^{M+N} \lambda_i^{req}} \tag{3.4}$$

The mean offered traffic associated with $Q_{i,j}$ at C_i is

$$\rho_{i,j}^{req} = \lambda_{i,j}^{req} \cdot \beta_{i,j}^{req} \tag{3.5}$$

The total offered traffic at C_i is

$$\rho_i^{req} = \sum_{j=1}^{N} \rho_{i,j}^{req} \tag{3.6}$$

5. The traffic generated at each S-node, S_j, $1 \le j \le N$ is due to responses. The cyclic server serves a queue of responses called Q_j. The first moment of the response arrival process is

$$\lambda_j^{resp} = \sum_{i=1}^{M+N} \lambda_{i,j}^{req} \tag{3.7}$$

Each response may generate one or more DTUs. Denote

$$\begin{aligned} b_j^{resp} &= \text{the number of DTUs per response for responses at } S_j \\ \beta_j^{resp} &= E[b_j^{resp}] \end{aligned} \tag{3.8}$$

The mean number of DTUs per response is

$$\beta^{resp} = \frac{\sum_{j=1}^{N} \lambda_j^{resp} \, \beta_j^{resp}}{\sum_{j=1}^{N} \lambda_j^{resp}} \tag{3.9}$$

The mean offered traffic at S_j is

$$\rho_j^{resp} = \lambda_j^{resp} \cdot \beta_j^{resp} \tag{3.10}$$

6. There are no error control mechanisms other than those built-in the request-response communication itself. A client entity sends a request and the response serves also as an implicit acknowledgement.
7. Each queue has an infinite buffer capacity. There are $N(N+M)$ queues at C-nodes and N queues at S-nodes.

a. An asymmetric system

Consider an asymmetric system with different arrival rates and average number of DTUs per request/response at each node. The global request-response arrival rate in the system is

$$\lambda = \sum_{i=1}^{M+N} \lambda_i^{req} + \sum_{j=1}^{N} \lambda_j^{resp} \tag{3.11}$$

Using (3.1) and then (3.7) λ can be expressed as

$$\lambda = \sum_{j=1}^{N} \left[\sum_{i=1}^{M+N} \lambda_{i,j}^{req} + \lambda_j^{resp} \right] = 2 \sum_{j=1}^{N} \lambda_j^{resp} \tag{3.12}$$

From (3.11) and (3.12) it follows that

$$\sum_{i=1}^{M+N} \lambda_i^{req} = \sum_{j=1}^{N} \lambda_j^{resp} = \frac{\lambda}{2} \tag{3.13}$$

This is a *flow conservation* relation which expresses the fact that any request generated at some C-node has associated with it a response provided by some S-node.

The offered traffic in the network is

$$\rho = \sum_{i=1}^{N+M} \rho_i^{req} + \sum_{j=1}^{N} \rho_i^{resp} \tag{3.14}$$

Define the mean number of messages in a request or response as

$$\beta = \sum_{i=1}^{N+M} \frac{\lambda_i^{req}}{\lambda} \beta_i^{req} + \sum_{j=1}^{N} \frac{\lambda_j^{resp}}{\lambda} \beta_j^{resp} \tag{3.15}$$

The second moments of the number of messages in a request or response are denoted by $\beta_i^{(2)req}$ and $\beta_j^{(2)resp}$. It follows that

$$\beta^{(2)} = \sum_{i=1}^{M+N} \frac{\lambda_i^{req}}{\lambda} \beta_i^{(2)req} + \sum_{j=1}^{N} \frac{\lambda_j^{resp}}{\lambda} \beta_i^{(2)resp} \tag{3.16}$$

The total traffic ρ can be expressed as (see 3.13)

$$\rho = \lambda \beta \tag{3.17}$$

b. A symmetric system

In this case all arrival rates at C-nodes are equal, all response rates at S-nodes are equal, the average number of DTUs in a request are the same, and the average number of DTUs in a response are the same.

The mean request arrival rates of all C-nodes are equal, therefore from (3.13) it follows that

$$\lambda_i^{req} = \lambda^{req} = \frac{\lambda}{2\,(M+N)} \quad \text{for } 1 \le i \le M+N \tag{3.18}$$

Similarly, the average response arrival rates of all S-nodes are equal and from (3.12) it follows that

$$\lambda_j^{resp} = \lambda^{resp} = \frac{\lambda}{2\,N} \quad \text{for } 1 \le j \le N \tag{3.19}$$

Consider now the offered traffic. The symmetry implies that all requests have the same average number of DTUs

$$\beta_{i,j}^{req} = \beta^{req} \quad \text{for } 1 \le i \le M+N, \text{ and } 1 \le j \le N \tag{3.20}$$

The offered traffic at C-node i due to requests becomes

$$\rho_i^{req} = \sum_{j=1}^{N} \rho_{i,j}^{req} = \beta^{req}\,\lambda^{req} \tag{3.21}$$

It follows from (3.18) that

$$\rho_i^{req} = \rho^{req} = \beta^{req}\,\frac{\lambda}{2(M+N)} \quad \text{for } 1 \le i \le M+N \tag{3.22}$$

Similarly, all responses have the same average number of DTUs

$$\beta_j^{resp} = \beta^{resp} \quad \text{for } 1 \le j \le N \tag{3.23}$$

From (3.19) it follows that

$$\rho_j^{resp} = \rho^{resp} = \beta^{resp}\,\frac{\lambda}{2\,N} \quad \text{for } 1 \le j \le N \tag{3.24}$$

The expressions for β and $\beta^{(2)}$ become (see 3.15 and 3.16)

$$\beta = \frac{\beta^{req} + \beta^{resp}}{2} \tag{3.25}$$

$$\beta^{(2)} = \frac{\beta^{(2)req} + \beta^{(2)resp}}{2} \tag{3.26}$$

3.2 Modeling assumptions - the remote execution

In the following it is assumed that at each S node S_n, $1 \le j \le N$ there is a server $SERV_j$. The delay analysis, namely the time spent by a request at $SERV_j$, and the ergodicity conditions associated with remote execution are of primary concern.

For simplicity a symmetric system is considered. The assumptions related to modeling of remote execution are:

(1) There are N identical servers $SERV_j$, $1 \le j \le N$, one located at each S-node of the network.

(2) The arrival process at each $SERV_j$ is a Poisson process with average arrival rate λ^{exec}. The discussion is restricted to a system in equilibrium. In this case the average arrival rate of requests at $SERV_j$ equals the average response rate at the corresponding Q_j, given by (3.19).

$$\lambda^{exec} = \lambda^{resp} = \frac{\lambda}{2N} \tag{3.27}$$

4. THE DELAY ANALYSIS

The delay analysis for a distributed system using request-response communication is presented in this section. The expected response time is the sum of the expected total communication time, T_{comm} and the expected remote execution time T_{rexec}

$$T = T_{comm} + T_{rexec} \tag{4.1}$$

In this expression the expected communication time is the sum of the expected queueing delays for communication, T^w_{comm}, and the expected transmission time and propagation delay for a request, T^{req} and for a response, T^{resp}

$$T_{comm} = T^w_{comm} + T^{req} + T^{resp}. \tag{4.2}$$

T^w_{comm} and T_{rexec} are evaluated separately in this section using the two independence assumptions presented earlier.

4.1 Application of Pseudo Conservation Laws to Communication Delay Analysis.

The pseudo conservations laws discovered by Watson [13] and extended by Boxma and Groenedijk [2] to allow a mixture of different service strategies at different queues, are generalizations of the work conservation principle of Kleinrock. Kleinrock [7] shows that "in a conservative system, weighted sums of the waiting times can never change no matter how sophisticated or elaborate the queuing discipline may be." In a conservative system no work is created or destroyed. The conservation law states that if the server is not idle then the service discipline may favor a class of customers over another but ultimately the amount of unfinished work in the system is the same. The pseudo conservation laws extends this principle to servers with vacation.

Weighted sums of the queuing delays in communication for a system using a mix of service disciplines can be expressed using the pseudo conservation laws. The system consists of $M + 2N$ queues as follows. There are N queues, one at each S-node. The N

queues $Q_{i,j}$ at any C-node C_i, $1 \le i \le N + M$ are aggregated into a single queue, Q_i, $1 \le i \le N + M$, with an aggregate mean arrival rate λ_i^{req}. The mean cycle time is derived and then the delay analysis of the equivalent system is carried out. Exhaustive service at all Q_j, $1 \le j \le N$ associated with server entities and at all Q_i, $1 \le i \le M + N$, associated with client entities is considered first. Then, the case of 1-limited service at queues associated with client entities is analyzed.

a. The cycle time analysis

Assume that the system is in equilibrium and call s_i the mean switch over time from the k^{th} to $(k + 1)$ queue, and $s^{(2)}$ the second moment of the switch over time. Then the first two moments of the total switch over time are

$$s = \sum_{k=1}^{M+2N} s_k \qquad (4.3)$$

and

$$s^{(2)} = \sum_{k=1}^{M+2N} s_k^{(2)}$$

Now the mean visit times at an S-queue and at a C-queue are respectively

$$E[V_j] = \rho_j^{resp} E[C] \text{ for } 1 \le j \le N \qquad (4.4)$$

and

$$E[V_i] = \rho_i^{req} E[C] \text{ for } 1 \le i \le M + N \qquad (4.5)$$

with $E[C]$ the mean cycle time, the mean interval between two consecutive visits of the cyclic server at a given queue. Summing over all visits for a cycle

$$\sum_{i=1}^{M+N} E[V_i] + \sum_{j=1}^{N} E[V_j] = E[C] - \sum_{k=1}^{M+2N} s_k$$

or

$$\sum_{i=1}^{M+N} \rho_i^{req} E[C] + \sum_{j=1}^{N} \rho_j^{resp} E[C] = E[C] - s$$

and

$$E[C] = \frac{s}{1 - \rho} \qquad (4.6)$$

with ρ given by expression (3.17). The expected visit time and intervisit time for the queues, associted with server entities are

$$E[V^{resp_j}] = \rho_j^{resp} \frac{s}{1 - \rho} \qquad (4.7)$$

$$E[I^{resp_j}] = E[C] - E[V^{resp_j}] = \frac{s}{1-\rho}\,[1 - \rho_j^{resp}] \qquad (4.8)$$

for $1 \le j \le N$

The corresponding visit and intervisit time for the queues associted with client entities are

$$E[V^{req_i}] = \rho_i^{req}\,\frac{s}{1-\rho} \qquad (4.9)$$

$$E[I^{req_i}] = E[C] - E[V^{req_i}] = \frac{s}{1-\rho}\,[1 - \rho_i^{req}] \qquad (4.10)$$

for $1 \le i \le N + M$.

For a symmetric system

$$E[V^{req_i}] = E[V^{req}] \text{ for } \ 1 \ \le i \le M + N \qquad (4.11)$$

From (3.22) and (4.9) it follows that

$$E[V^{req}] = \beta^{req}\,\frac{\lambda}{2(M+N)}\,\frac{s}{1-\lambda\beta} \qquad (4.12)$$

Then the average number of requests arriving at queue Q_i located at C-node C_i during a cycle time is

$$E[N^{req}] = \frac{\lambda}{2(M+N)}\,\frac{s}{1-\lambda\beta} \qquad (4.13)$$

Similarly from (3.22) and (4.7)

$$E[V^{resp}] = \beta^{resp}\,\frac{\lambda}{2N}\,\frac{s}{1-\lambda\beta} \qquad (4.14)$$

The average number of responses arriving at queue Q_j located at S-node C_j during a cycle time is

$$E[N^{resp}] = \frac{\lambda}{2N}\,\frac{s}{1-\lambda\beta} \qquad (4.15)$$

b. The delay analysis when the service discipline is exhaustive at all queues.

The service discipline in case of a server with vacation (cyclic server), defines the policy used by the server when visiting a certain queue. A *1-limited* service discipline means that only one request is processed when the server visits the queue. *Exhaustive* service means that all requests are processed. Other service disciplines are *gated*, when the server processes all requests in the queue at the time of its arrival, and *semi-gated* when the server leaves in the queue one less that the number at the time of its arrival. Following [2]

$$\sum_{i=1}^{M+N} \rho_i^{req}\, EW_i^{req} + \sum_{j=1}^{N} \rho_j^{resp}\, EW_j^{resp} =$$

$$\frac{\lambda\beta^{(2)}}{2(1-\rho)}\,\rho + \frac{(\lambda^{(2)} - \lambda^2 - \lambda)\beta}{2\lambda(1-\rho)}\,\rho + \rho\,\frac{s^{(2)}}{2s} - \frac{1}{2}\,\rho \tag{4.16}$$

$$+ \frac{s}{2(1-\rho)}\left[\rho^2 - \sum_{i=1}^{M+N} (\rho_i^{req})^2 - \sum_{j=1}^{N} (\rho_j^{resp})^2\right]$$

EW_i^{req} is the mean waiting time of a request arriving at a C-node C_i, and EW_j^{resp} is the mean waiting time of a response generated at an S-node S_j.

In case of a symmetric system the following expression is obtained

$$EW_i^{req} = EW^{req} \qquad 1 \le i \le M+N \tag{4.17}$$

$$EW_j^{resp} = EW^{resp} \qquad 1 \le j \le N \tag{4.18}$$

$$\rho^2 = \lambda^2\left[\frac{\beta^{req} + \beta^{resp}}{2}\right]^2 \tag{4.19}$$

When N and $N + M$ are large the following approximation is justified

$$\sum_{i=1}^{M+N} (\rho_i^{req})^2 + \sum_{j=1}^{N} (\rho_j^{resp})^2 = (M+N)\left[\frac{\beta^{req}\lambda}{2(M+N)}\right]^2 + N\left[\frac{\beta^{resp}\lambda}{2N}\right]^2 \ll \rho^2 \tag{4.20}$$

With this approximation (4.16) becomes

$$\frac{\beta^{req}}{\beta} EW^{req} + \frac{\beta^{resp}}{\beta} EW^{resp} = \tag{4.21}$$

$$= \frac{1}{(1-\lambda\beta)}\left[\lambda\beta^{(2)} + \frac{\lambda^{(2)}}{\lambda}\beta - \lambda\beta - \beta + s\lambda\beta\right] + \left[\frac{s^{(2)}}{s} - 1\right]$$

When $\beta^{req} = \beta^{resp}$ the average queuing delay for request-response communication is given by

$$T^{w}_{comm} = EW^{req} + EW^{resp} = \frac{1}{(1-\rho)}\left[\beta c_{\lambda} + \rho c_{\beta} + s\rho\right] + c_s \qquad (4.22)$$

The following notation is used for random variables λ, β and s

$$c_x = \frac{x^{(2)}}{x} - 1 \qquad (4.23)$$

with x and $x^{(2)}$ the first and the second moments of random variable X.

c. The delay analysis with different service disciplines.

Consider now the case when the service discipline is 1-limited for queues associated with client entities and exhaustive for the ones associated with server entities. The results from [2] can be applied to obtain

$$\sum_{i=1}^{M+N} \rho_i^{req}\left[1 - \frac{\lambda_i^{req}\cdot s}{1-\rho}\right] EW_i^{req} + \sum_{j=1}^{N} \rho_j^{resp}\, EW_j^{resp} =$$

$$\frac{\lambda\beta^{(2)}}{2(1-\rho)}\,\rho + \frac{(\lambda^{(2)} - \lambda^2 - \lambda)\beta}{2\lambda(1-\rho)}\,\rho + \rho\,\frac{s^{(2)}}{2s} - \frac{1}{2}\,\rho + \qquad (4.24)$$

$$+ \frac{s}{2(1-\rho)}\left[\rho^2 + \sum_{i=1}^{M+N}(\rho_i^{req})^2 + \sum_{j=1}^{N}(\rho_j^{resp})^2 + 2\sum_{i=1}^{M+N}\frac{(\lambda_i^{req})^{(2)} - \lambda_i^{req}}{2\lambda_i^{req}}\,\rho_i^{req}\right]$$

The case of a symmetric system is considered now. The approximation discussed previously, (4.20) is used. In addition

$$2\sum_{i=1}^{M+N}\frac{(\lambda_i^{req})^{(2)} - \lambda_i^{req}}{2\lambda_i^{req}}\,\rho_i^{req} = \lambda\,\beta^{req}\,c_{\lambda} \qquad (4.25)$$

With these approximations the expression (4.16) becomes

$$\frac{\beta^{req}}{\beta}\left[1 - \frac{\lambda^{req}\cdot s}{1-\rho}\right]EW^{req} + \frac{\beta^{resp}}{\beta}EW^{resp} = \qquad (4.26)$$

$$= \frac{\beta}{(1-\lambda\beta)}\left[\lambda\,(c_{\beta} + s) + c_{\lambda} + 1 - \lambda^2\right] + \lambda\beta c_s + \frac{s}{1-\rho}\lambda\,\beta^{req}\,c_{\lambda}$$

4.2 The Analysis of the Remote Execution.

Consider an M/G/1 queue [7] with

- λ^{rexec} the first moment of the request arrival rate,
- μ^{rexec} and $\mu^{rexec,(2)}$ the first and the second moments of the request service rate,
- $\rho^{rexec} = \lambda^{rexec} / \mu^{rexec}$.

Then the Polaczek-Khinchin formula gives the total time in the system as

$$T_{rexec} = \mu^{rexec} + \frac{\lambda^{rexec}\ \mu^{rexec,(2)}}{2\,(1 - \rho^{rexec})}. \tag{4.27}$$

Clearly, when the service process is Poisson the time in the system is given by

$$T_{rexec} = \frac{1}{\mu^{rexec}\ (1 - \rho^{rexec})}. \tag{4.28}$$

5. THE ANALYSIS OF SIMULATION RESULTS

The approximate delay analysis presented in this paper is based upon two assumptions, the remote execution independence assumption and the communication independence assumption. These assumptions must be supported by empirical evidence. The first step in gathering empirical evidence is through simulation. Further confirmation through measurements in an experimental system are needed.

The results of a simulation experiment designed to determine the correctness of the approximations discuused above are presented. *A homogeneous systems* with computation load spread evenly among all servers is considered. The system is said to be *communication-bound* if the average channel utilization is significantly larger than the average server utilization. A system with average channel utilization close to the average server utilization is called a *balanced system.* The request rate is determined by the load placed upon the system. High request rates are associated with small DTU size for requests/responses and with short expected remote execution time. Large DTU size and significant remote execution time are associated with low request rates. For the purpose of analysis three types of transaction oriented systems are defined. *Type A* corresponds to a communication-bound system with relatively high request arrival rate. *Type B* corresponds to a balanced system with a high request rate and *Type C* corresponds to a communication bound system with low request rate. Three types of traffic are considered, *light traffic*, when channel utilization is lower than 55%, *medium traffic* when channel utilization does not exceed 75%, and *heavy traffic* with channel utilization higher than 75%. In all models the average DTU size for a request and a response are equal.

The first objective of the simulation experiment is to compare two systems, one called *Model 1* uses request-response communication and the second one called *Model 2* is an equivalent system with independent arrival rates at all nodes. The two systems have the same number of nodes and the average traffic load is the same. The DTU size is exponentially distributed. Moreover, the average message arrival rate at the corresponding nodes of Model 2, the one with request-response communication and Model 1, the system with independent arrival processes at all nodes, are the same. The average traffic associated with requests equals the traffic associated with responses for

Model 2. The dependence of the cycle time and of the average queueing delay for transmission for the two models are studied. The effects of: the DTU size distribution, the service discipline, the ratio of client to server nodes, and the traffic load are investigated.

Figures 1, 2 and 3 present the results for a Type A transaction oriented system for a system with exhaustive service at client and server nodes, for a system with exhaustive service at server nodes and 1-limited at client nodes, and for a system with 1-limited service discipline at all nodes respectively. The results are plotted for three different client to server node ratios, 90:10, in Figures 1a, 2a, and 3a, 80:20 in Figures 1b, 2b and 3b, and 60:30 in Figures 1c and 2c. The following conclusions may be drawn from theses data. The relative errors are lower for the cycle time than for the queuing delays when Model 1 is used to approximate the communication delays of Model 2. The approximation is better for light traffic. The relative errors are less than 1% for the cycle time and less than 1.1% for the queueing delays in case of light traffic. The upper limits for the errors increase to 2.5% and 3% respectively at medium traffic, $U_{ch} \leq 75\%$, and reach 6% for heavy traffic. The effect of client to server node ratio has little impact upon the results. The effect of the service discipline is important and the approximations are best suited for the case of exhaustive service discipline at all nodes. The use of 1-limited service discipline at client nodes does not affect the results in any significant way but the 1-limited service discipline at server nodes leads to larger relative errors. For example the relative errors increase up to 8% at medium traffic load.

In case of a Type B system the results are essentially the same as those reported for a Type A system and the corresponding graphs are omitted. Figures 9, 10 and 11 present the results for a Type C system because a new effect is observed. The average queueing delay is larger than the average cycle time and increases rapidly with the traffic load (or channel utilization). The average DTU size is fairly large in case of a Type C system, (20 times larger than in case of Type A or B systems), and the request arrival rate for a given channel utilization is correspondingly lower for a Type C system. As a result, there are many cycles with no traffic, but whenever a node transmits, there is a large chance that other nodes will receive requests that have to be queued waiting for the availability of the channel. In case of Type A or B systems the average queueing delay is always shorter than the average cycle time, as shown in Figures 1, 2 and 3. A slight increase in errors introduced by the approximations discussed above can be observed. For example in case of exhaustive service at client and server nodes, Figure 9, the errors in delay at medium traffic load increase from 3 to 5%.

The analysis of the data obtained from the simulation experiments described above leads to the conclusion that *the independence assumption is quite satisfactory for the case of a token ring network with light to medium traffic*, when the individual request arrivals can be modeled as independent Poisson streams, and when the DTU size has exponential distributions. For exhaustive service discipline at all nodes, or exhaustive at server nodes and 1-limited at client nodes, the approximation leads to a relative error of at most 5% for traffic up to 75% of the channel capacity.

A second objective of this study is to determine if a remote server can be modeled as an M|M|1 system in case of exponentially distributed remote service time for Model 2. The results of a comparison of the average queuing delays predicted by an M|M|1 approximation with the average queuing delays observed in the simulation of Type A, B, and C transaction oriented systems are presented in Figures 4, 7 and 12 respectively. The client to server node ratio is 90:10 because such a ratio is expected in systems of practical interest. *The relative error when the queuing delays at a remote server are approximated by an M|M|1 model is not larger than 6% for any of the three types of systems discussed in this paper, for server utilizations up to 75%.* In general, the exhaustive service discipline in communication, at all nodes, leads to the lowest queuing delay at the remote server. The results presented above indicate that the second independence assumption is well justified and fairly robust.

Finally, the expected response time predicted by the approximation introduced in this paper is compared with the response time observed in simulation. The results for Type A and B systems are presented in Figures 5 and 8. The results for a Type C system are quite similar and are omitted. The approximation discussed in this paper is inaccurate for server utilization lower than 5% (the corresponding channel utilization is about 10% for a type A system and 5% for a Type B system), but it is fairly accurate for server utilization in the 20% to 80% range, where it leads to relative errors of less than 7%. Figure 6 provides an explanation for the inaccuracy at low load. The approximations of the queuing delays in communication derived from the pseudo conservation laws are not acceptable at low traffic load. Other approximations like the ones in [6] fit much better the simulation results and can be used in case of exhaustive service strategy at all nodes.

The errors reported in this paper are upper limits for errors observed when Model 1 is used to approximate the behavior of Model 2 in repeated simulation experiments, typically 5 to 10 runs for every point. To reach steady-state a typical simulation experiment needs close to 50 hours of CPU time on a SUN 3/50 workstation. The entire project used a network of 10 workstations for more than six weeks.

CONCLUSIONS

The response time in transaction oriented communication, also called request-response communication, consists of the queueing delay, the transmission time and the propagation delay for the transmission of both request and response, and the queueing delay and the service time at the remote server. An exact delay analysis for such a system is extremely hard for two reasons. First, a salient feature of all models of computer networks is the independence assumption, the arrival process at each node of the network is independent of the arrival process at all other nodes. Second, the arrival process at a server entity depends upon the Medium Access Control method used in the network. Approximations for the response time in a transaction oriented system are investigated in this paper. The method based upon this approximation can be formulated as follows:

(1) Determine the offered traffic load due to requests and responses and approximate the queueing delay in communication as if the arrival processes at all nodes are independent.

(2) Determine the computation load for each server. Estimate the request arrival rate and the service time and then compute the time in system at the remote server as predicted by an M|M|1 or an M|G|1 model depending upon the distribution of the remote service time.

The analysis presented in this paper covers only the case of a transaction oriented system built around a token ring. The pseudo-conservation laws are used to estimate queueing delays in data transmissions for different service strategies at client and server nodes in a heterogeneous network. The pseudo-conservation laws can be applied only if independent arrival processes at the nodes of a network can be assumed. Then queueing delays at the remote server modeled as an M|M|1 system are approximated. A simulation experiment to determine the range of systems parameters when the approximations are well justified is presented. The cycle time and the delay for two models are compared to verify assumption (A1). Model 1 corresponds to a system with independent arrival rates at all nodes and Model 2 to the equivalent system with transaction oriented communication. Two systems are said to be *equivalent* if they use the same Medium Access Control method and if the average traffic offered at each node of one model equals the traffic at a corresponding note of the second model. The relative error when the Model 1 is used to approximate Model 2 is not larger than 5%, for the exhaustive service strategy at client and server nodes, when channel utilization does not exceed 75%. To verify assumption (A2) the queueing delay at the remote server observed in the simulation are compared with the one predicted by the M|M|1 model. The relative errors are not larger than 6% for the case of exhaustive service at all nodes.

The response time predicted by the analytical model and the one observed in simulation are compared. The approximations are satisfactory for server utilization in the 20% to 80% range, for exhaustive service strategy at the client and the server nodes. The approximations are justified for relatively dense networks with light to medium traffic, (up to 75% of channel capacity), when the individual request arrival can be modeled as independent Poisson streams, and when the DTU sizes are exponentially distributed. These conditions are generally satisfied by distributed systems of practical interest. Further research to investigate the validity of the approximations suggested in this paper to other Medium Access Control methods are necessary.

ACKNOWLEDGMENTS

We want to express our thanks to Dr. Micha Hofri for a careful reading of an earlier version of the manuscript and to Dr. Herb Schwetman who has made available to us the CSIM simulation package used in this paper. Special thanks to one of the referees for his comments and suggestions. Srinivasan Jagannathan has written the simulation program.

LITERATURE

[1] Birell, A.D. and Nelson, B.J., "Implementing remote procedure calls", *ACM Trans. on Comp. Sys.*, Vol. 2, No. 1, pp. 39–59 (1984).

[2] Boxma, O.J. and Groenedijk, W.P., "Waiting times in discrete-time cyclic-server systems", *IEEE Trans. on Communications.*, Vol. 36, No. 2, pp. 164–171 (1988).

[3] Cheriton, D.R., "Request-response and multicast interprocess communication in the V kernel", in *Networking in Open Systems, Lecture Notes in Computer Science,* Vol. 248, pp. 296–312 (1986).

[4] Cheriton, D.R., "VMTP: versatile message transaction protocol – protocol specification", *RFC 1045,* (1988).

[5] Eberle, H.M. and Schmutz, H., "Network operating system kernels for heterogeneous environments", in *Networking in Open Systems, Lecture Notes in Computer Science,* Vol. 248, pp. 270–295 (1986).

[6] Hammond, J.L. and O'Reilley, P.J.P., " Performance analysis of local computer networks", Addison Wesley (1986).

[7] Kleinrock, L., "Queueing systems", Vol. 1, and Vol. 2, Wiley (1976).

[8] Marinescu, D.C., "Modeling programs with remote procedures", *Proc. 7th Inter. Conf. on Dist. Comp. Sys.,* Berlin, Computer Society Press, pp. 98–104 (1987).

[9] Marinescu, D.C., "Scheduled and nonscheduled access in a distributed system based upon a functional communication model", *8th Inter. Conf. on Dist. Comp. Sys.,* San Jose, Computer Society Press, pp. 197–205 (1988).

[10] Reiser, M., "Communication system models embedded in the OSI - reference model, a survey", in *Computer Networking and Performance Evaluation,* (T. Hasegawa, H. Takagi and Y. Takahashi, eds.), North-Holland, pp. 85–112 (1986).

[11] Sevcik, K.C. and Johnson, M.J., "Cycle time properties of the FDDI token ring protocol", *IEEE Trans. on Soft. Eng.,* Vol. SE-13, pp. 376–385 (1987).

[12] Takagi, H., "Analysis of polling systems", *MIT Press* (1986).

[13] Watson, K.S., Performance evaluation of cyclic server strategies – a survey", in *PERFORM'84,* (E. Gelenbe, ed.), North-Holland, pp. 521–533 (1984).

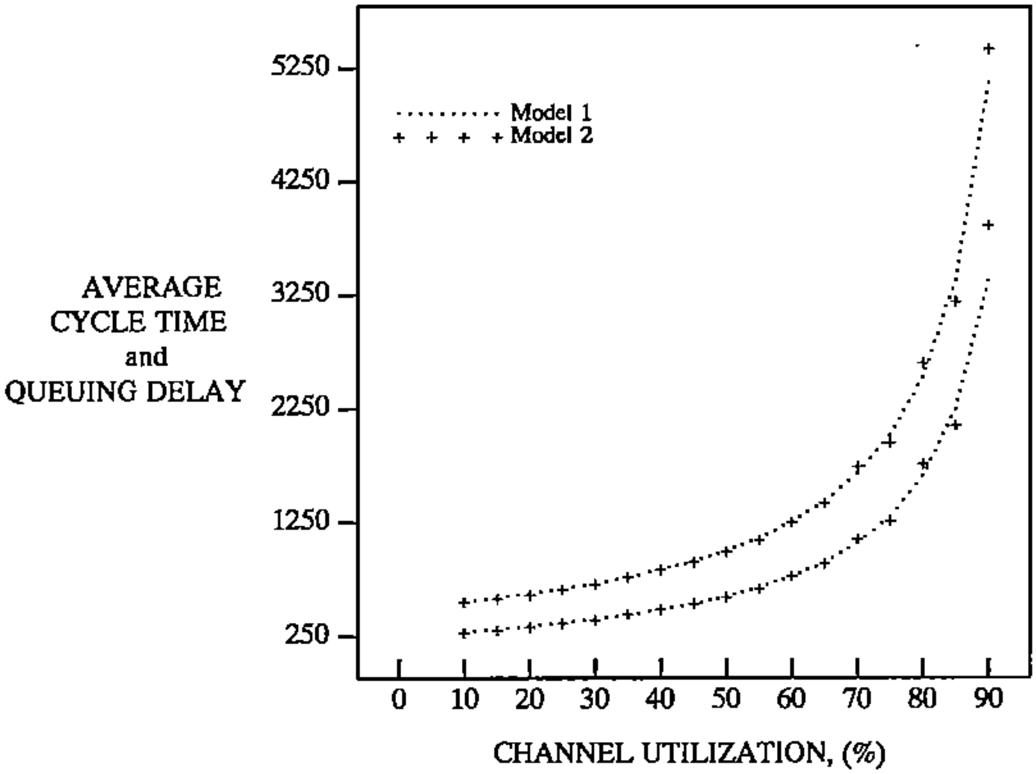

Figure 1a.

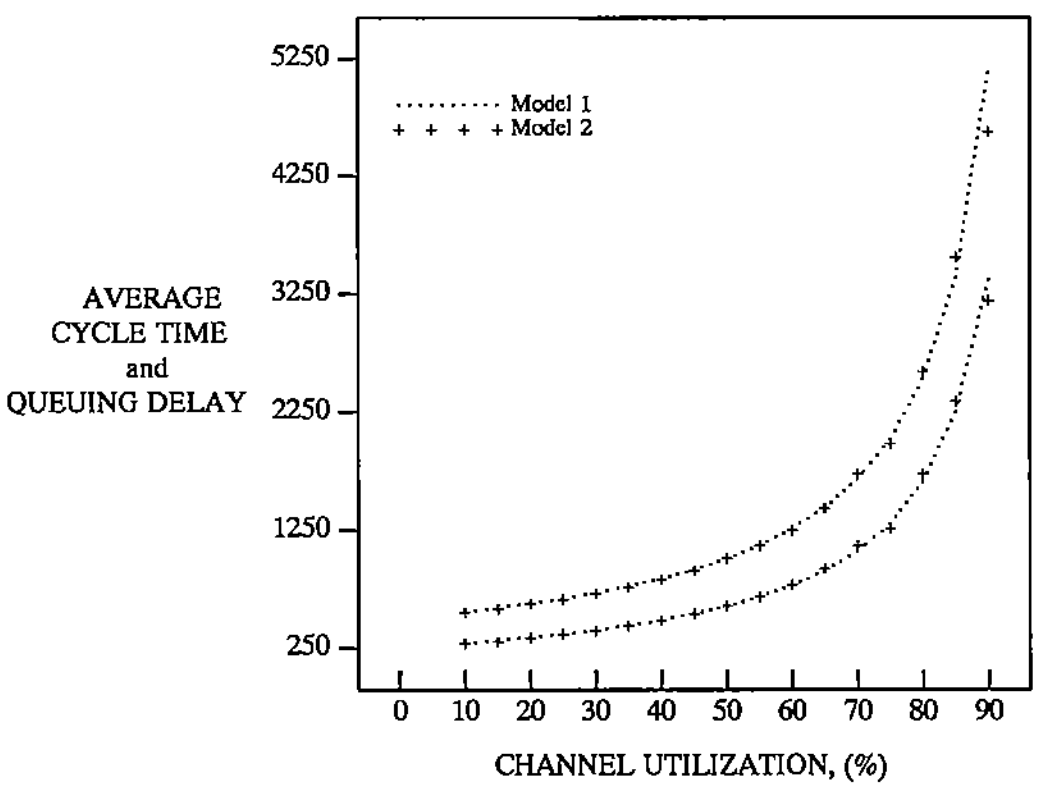

Figure 1b.

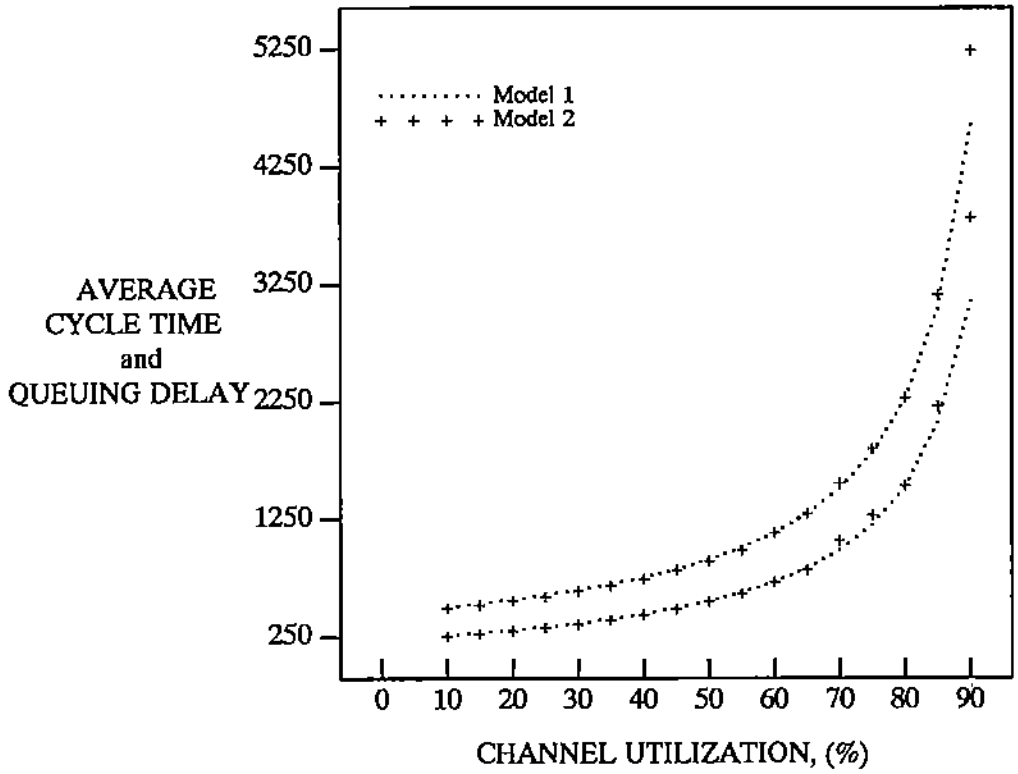

Figure 1c.

Figure 1. The average cycle time (upper curves) and the average queuing delay for communication (lower curves) versus channel utilization, U_{ch}. *Transaction-oriented system of type A*, large request arrival rate (average DTU size is 100 and average remote execution time 1000), and a communication bound system (average channel utilization twice the average server utilization). *Exhaustive service strategy at client and server queues.*

The system with request-response communication, Model 2 and the equivalent system with independent arrival rates at all nodes, Model 1. The system has a client to server ratio of: 90:10 (Figure 1a), 80:20 (Figure 1b) and 60:30 (Figure 1c).

The relative error in the cycle time when Model 1 is used to approximate the behavior of a request-response system, Model 2, is smaller than 1% for $U_{ch} \leq 55\%$ and is smaller than 2.5% for $U_{ch} \leq 75\%$. The corresponding upper limits for errors in queuing delays are 1.1% and 3%.

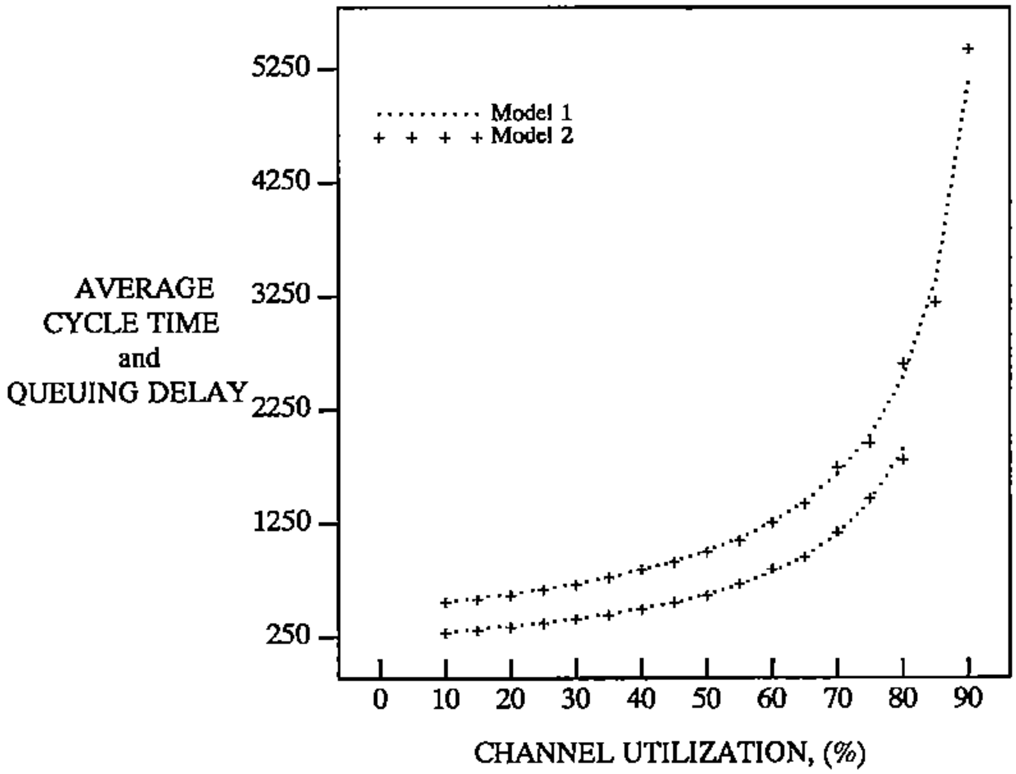

Figure 2a.

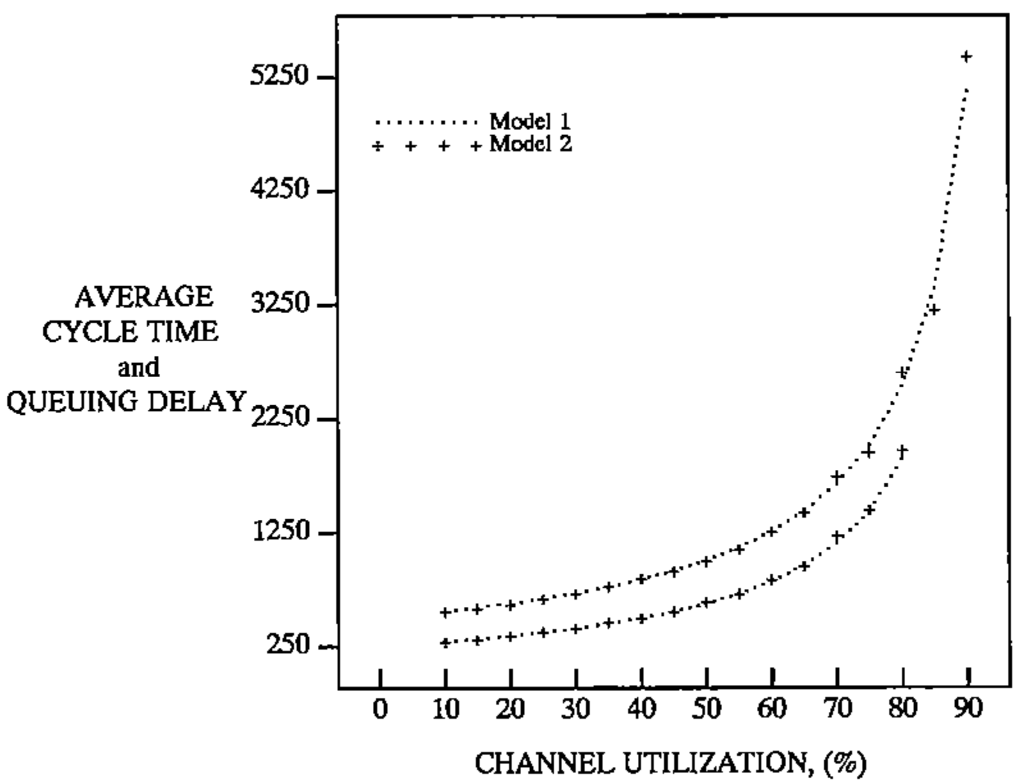

Figure 2b.

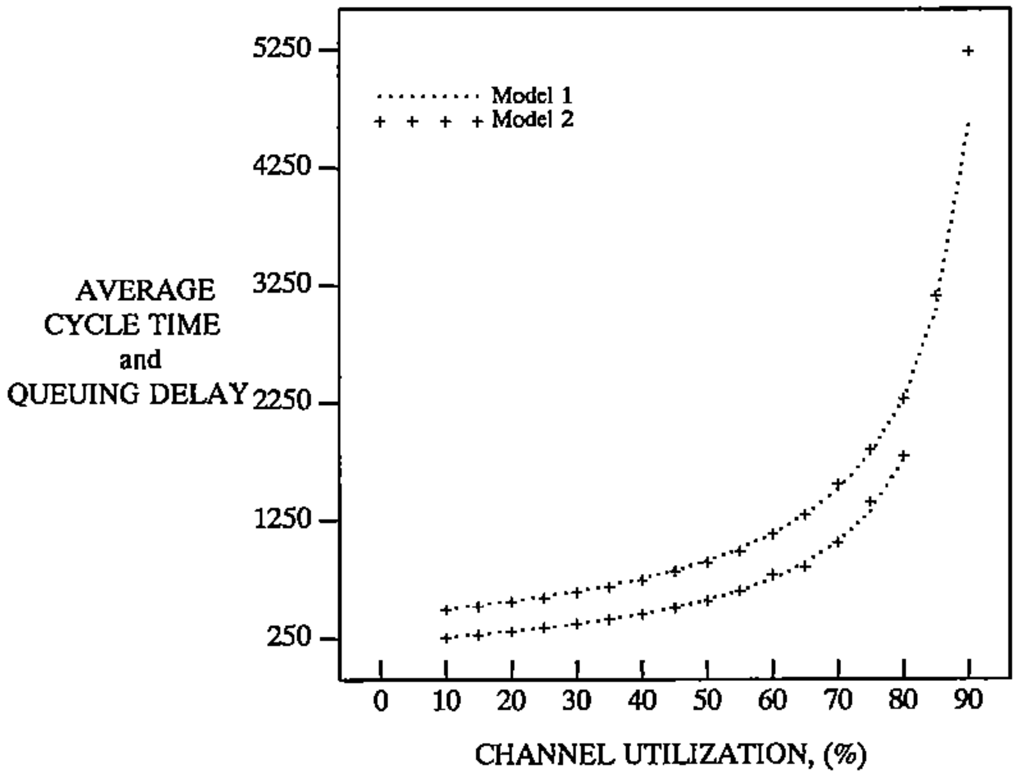

Figure 2c.

Figure 2. The average cycle time (upper curves) and the average queuing delay for communication (lower curves) versus channel utilization, U_{ch}. *Transaction-oriented system of type A*, large request arrival rate (average DTU size is 100 and average remote execution time 1000), and a communication bound system (average channel utilization twice the average server utilization). *Exhaustive service strategy at server queues and 1-limited at client queues.*

The system with request-response communication, Model 2, and the equivalent system with independent arrival rates at all nodes, Model 1. The system has a client to server ratio of: 90:10 (Figure 2a), 80:20 (Figure 2b) and 60:30 (Figure 2c).

The relative error in the cycle time when Model 1 is used to approximate the behavior of a request-response system, (Model 2, is smaller than 1% for $U_{ch} \leq 55\%$ and is smaller than 2.5% for $U_{ch} \leq 75\%$. The corresponding bounds for errors in queuing delays are 1% and 3%.

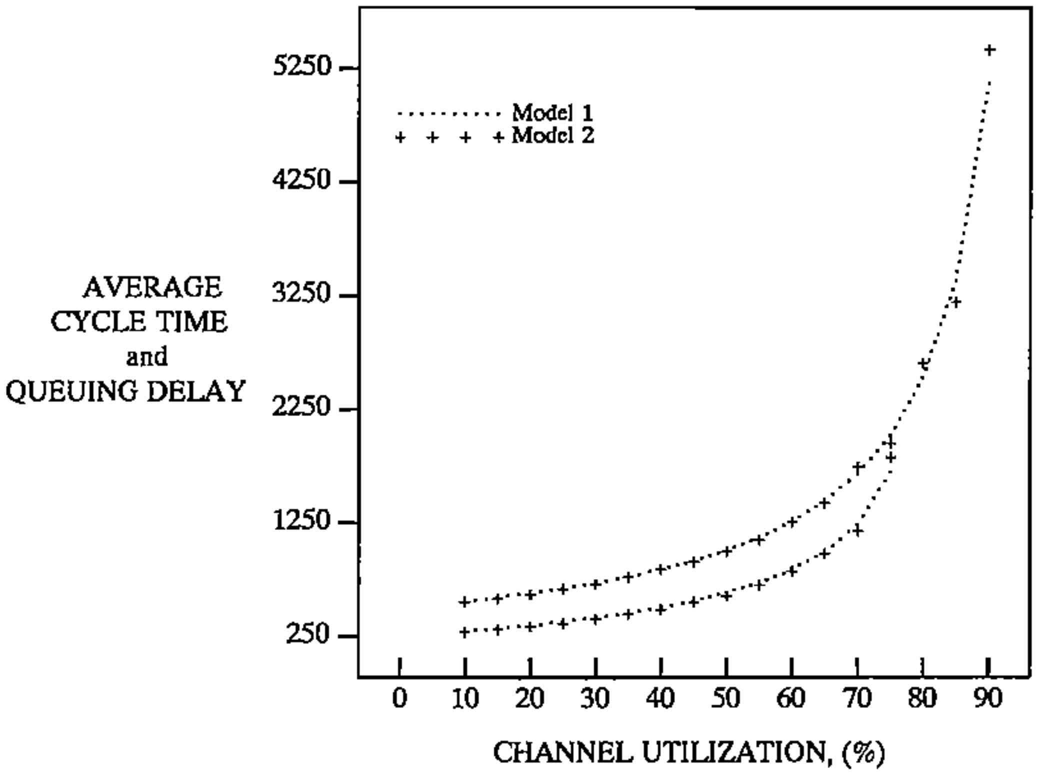

Figure 3a.

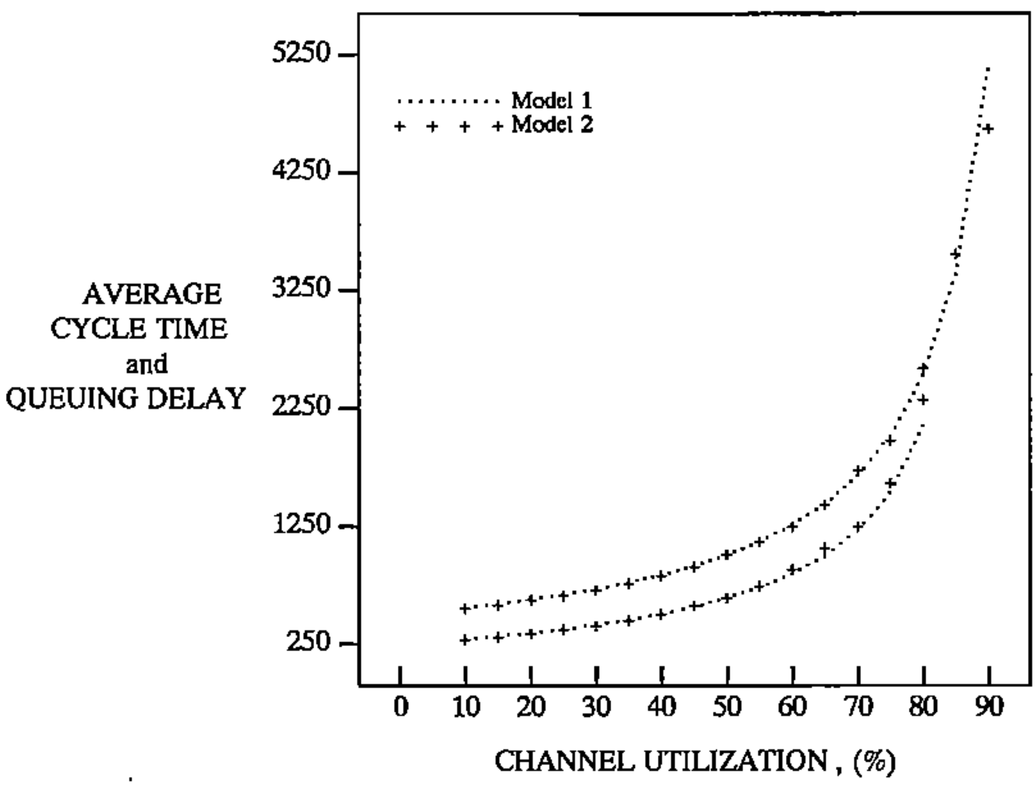

Figure 3b.

Figure 3. The average cycle time (upper curves) and the average queuing delay for communication (lower curves) versus channel utilization, U_{ch}. *Transaction-oriented system of type A*, large request arrival rate (average DTU size is 100 and average remote execution time 1000), and a communication bound system (average channel utilization twice the average server utilization). *1-limited service strategy at server queues and at client queues*

The system with request-response communication, Model 2 and the equivalent system with independent arrival rates at all nodes, Model 1. The system has a client to server ratio of: 90:10 (Figure 3a) and 80:20 (Figure 3b).

The relative error in the cycle time when Model 1 is used to approximate the behavior of a request-response system, Model 2, is smaller than 1% for $U_{ch} \leq 55\%$ and is smaller than 4.0% for $U_{ch} \leq 75\%$. The corresponding upper limits for errors in queuing delays are 3.0% and 8%.

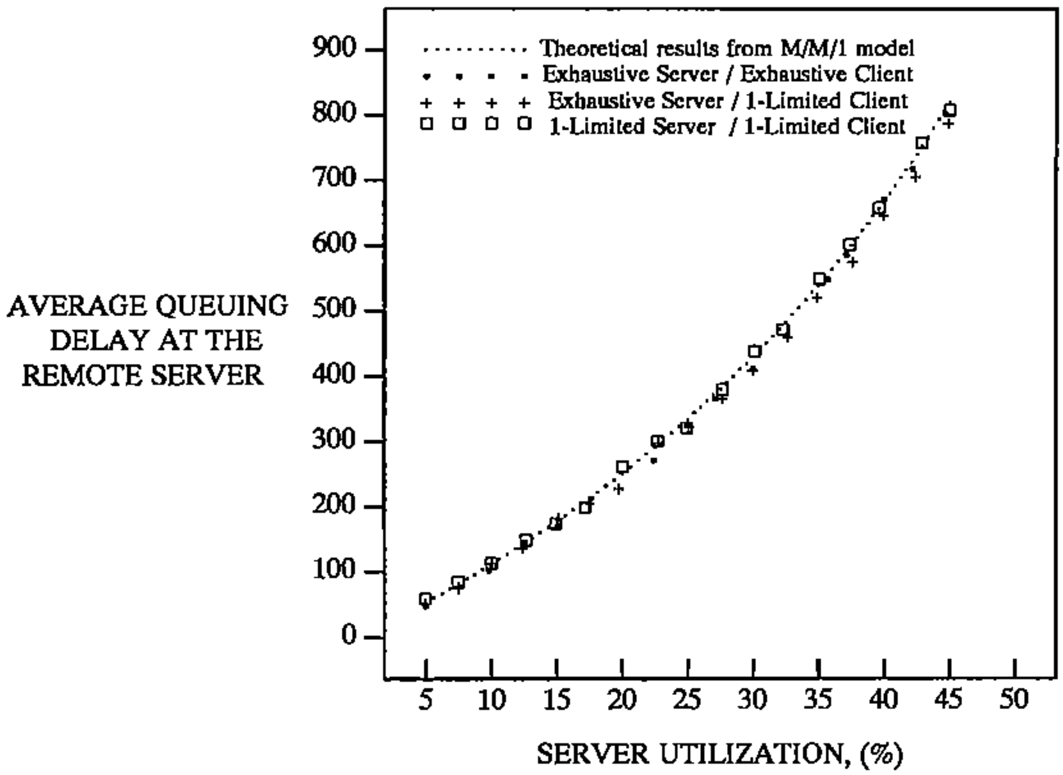

Figure 4. The average queuing delay at the remote server site function of the average server utilization. *Transaction-oriented system of type A*, large request arrival rate (average DTU size is 100 and average remote execution time 1000), and a communication bound system (average channel utilization twice the average server utilization). The computation load is spread evenly among all servers.

Four curves are plotted: the queuing delay as predicted by the M/M/1 system, the queuing delay when an exhaustive service strategy is used for communication in the ring at server and at client nodes, E/E case, the one when exhaustive service is replaced by 1-limited service at all transmission queues, $1/1$ case and finally when an exhaustive strategy is used at server nodes and a 1-limited at client nodes, the $E/1$ case. Client to server ratio is 90:10. The E/E case is best approximated by the M/M/1 model, 2% is an upper bound for the relative error in this approximation. In the $E/1$ case the largest error is 2.5%.

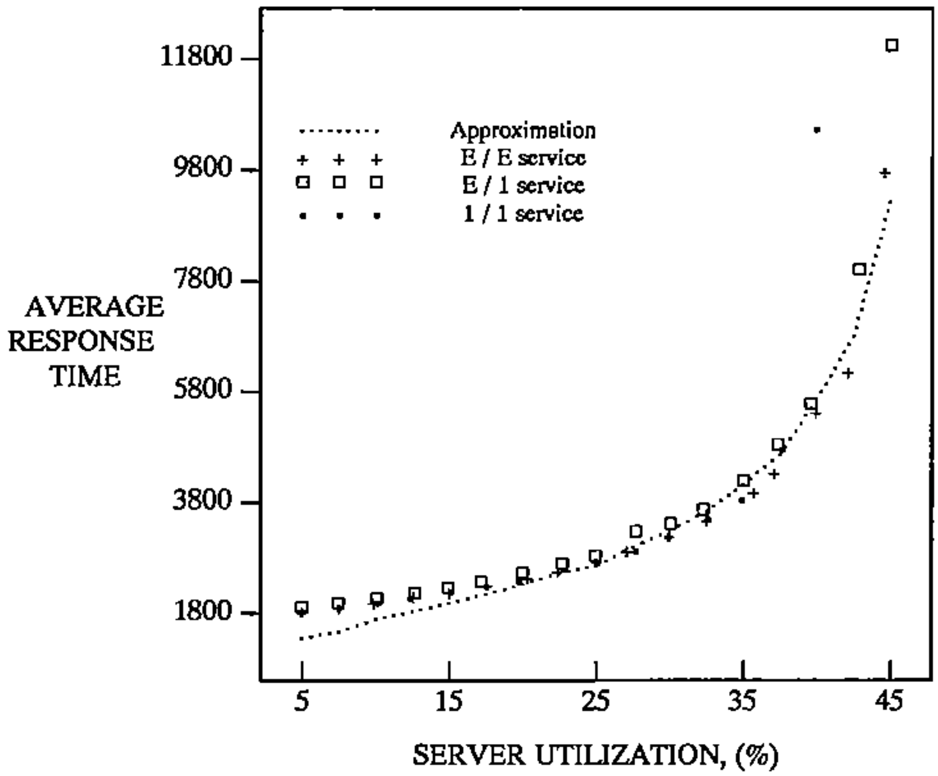

Figure 5. The average response time function of the average server utilization. *Transaction-oriented system of type A*, large request arrival rate (average DTU size is 100 and average remote execution time 1000), and a communication bound system (average channel utilization twice the average server utilization). The computation load is spread evenly among all servers.

Four curves are plotted: the response time predicted by the approximate analysis, the response time when an exhaustive service strategy is used for communication in the ring at server and at client nodes, E/E, the one when exhaustive service is replaced by 1-limited service at all transmission queues 1/1, and finally when an exhaustive strategy is used at server nodes and a 1-limited at client nodes, E/1. Client to server ratio is 90:10.

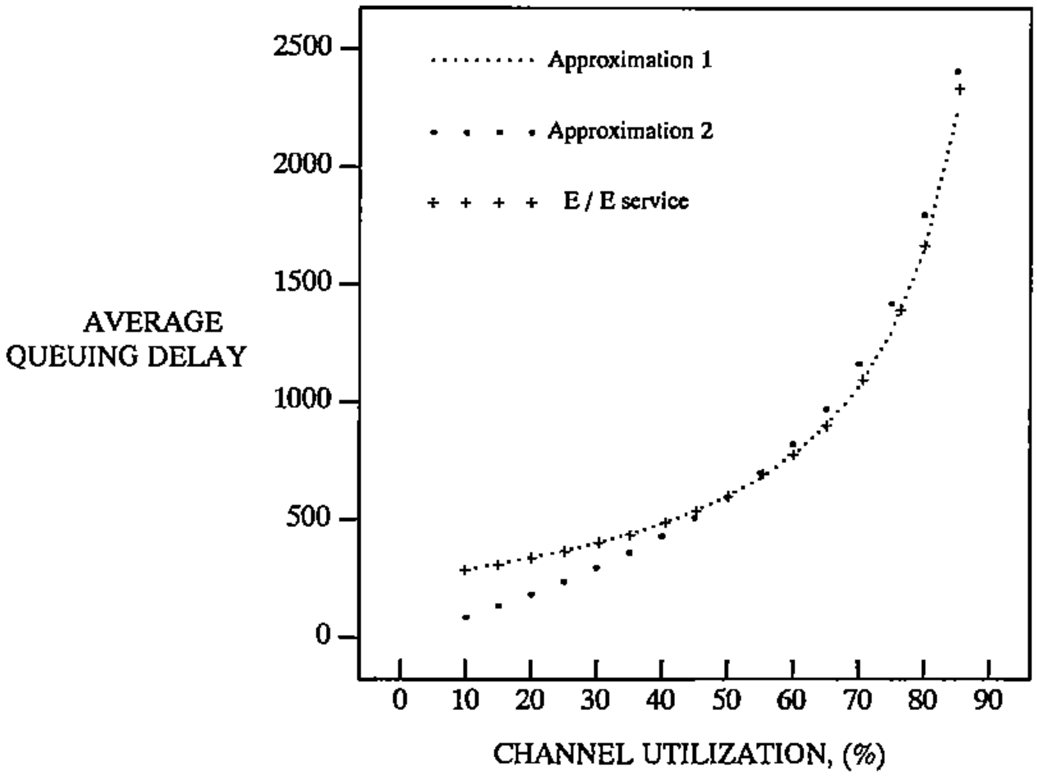

Figure 6. The average queuing delay in communication function of the average channel utilization. *Transaction-oriented system of type B,* large request arrival rate (average DTU size is 100 and average remote execution time 2000), and a balanced system, the channel utilization for this system with computation load spread evenly among all servers is within 5% of the average server utilization. An exhaustive service strategy is used for communication in the ring at server and at client nodes, *E/E*.

Three curves are plotted: the queuing delay predicted by the approximation given in [6] (Approximation 1), the one predicted by formula 4.22, (Approximation 2), and the queuing delay observed in simulation. Client to server ratio is 90:10.

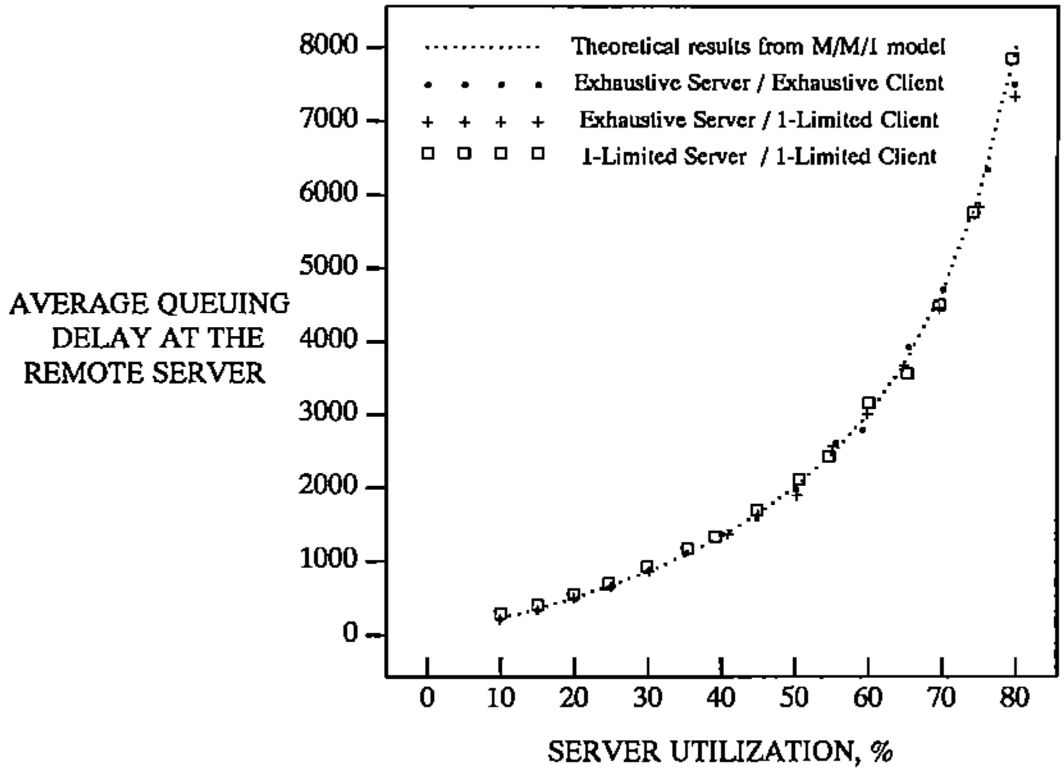

Figure 7. The average queuing delay at the remote server site function of the average server utilization. *Transaction-oriented system of type B*, large request arrival rate (average DTU size is 100 and average remote execution time 2000), and a balanced system, the channel utilization for this system with computation load spread evenly among all servers is within 5% of the average server utilization.

Four curves are plotted: the queuing delay as predicted by the M/M/1 system, the queuing delay when an exhaustive service strategy is used for communication in the ring at server and at client nodes, E/E case, the one when exhaustive service is replaced by 1-limited service at all transmission queues, the $1/1$ case and finally when an exhaustive strategy is used at server nodes and a 1-limited at client nodes, the $E/1$ case. Client to server ratio is 90:10.

The largest relative error observed for the E/E case is 6%, it increases to 8% for the $E/1$ case and is about 2.5% for the $1/1$ case.

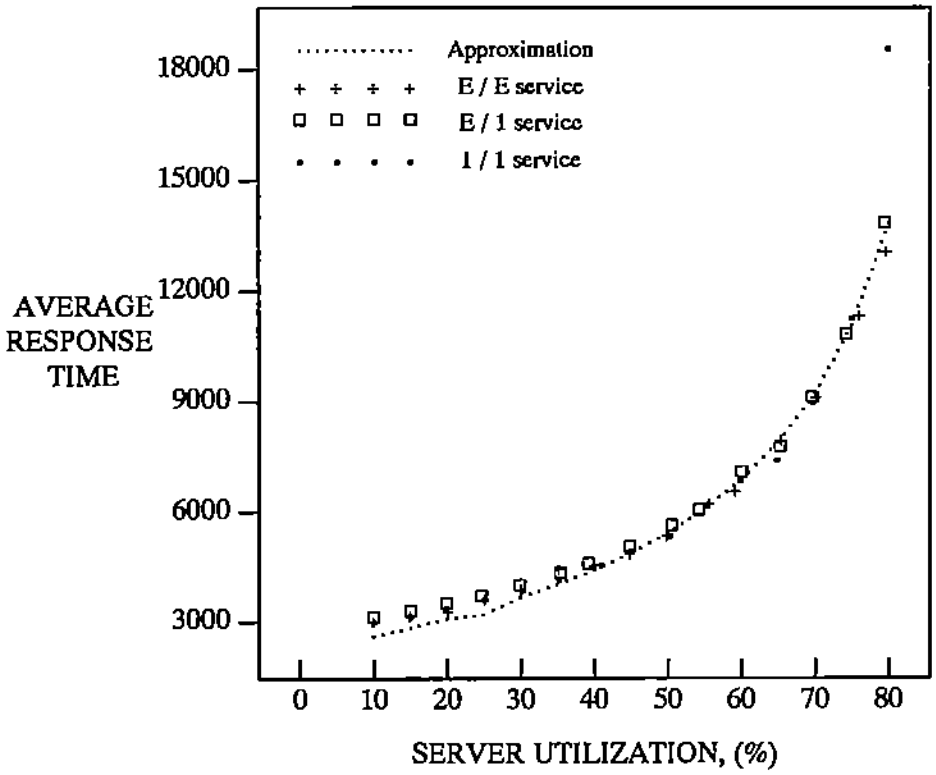

Figure 8. The average response time function of the average server utilization. *Transaction-oriented system of type B*, large request arrival rate (average DTU size is 100 and average remote execution time 2000), and a balanced system, the channel utilization for this system with computation load spread evenly among all servers is within 5% of the average server utilization.

Four curves are plotted: the response time predicted by the approximation, the response time when an exhaustive service strategy is used for communication in the ring at server and at client nodes, E/E, the one when exhaustive service is replaced by 1-limited service at all transmission queues 1/1, and finally when an exhaustive strategy is used at server nodes and a 1-limited at client nodes, E/1. Client to server ratio is 90:10.

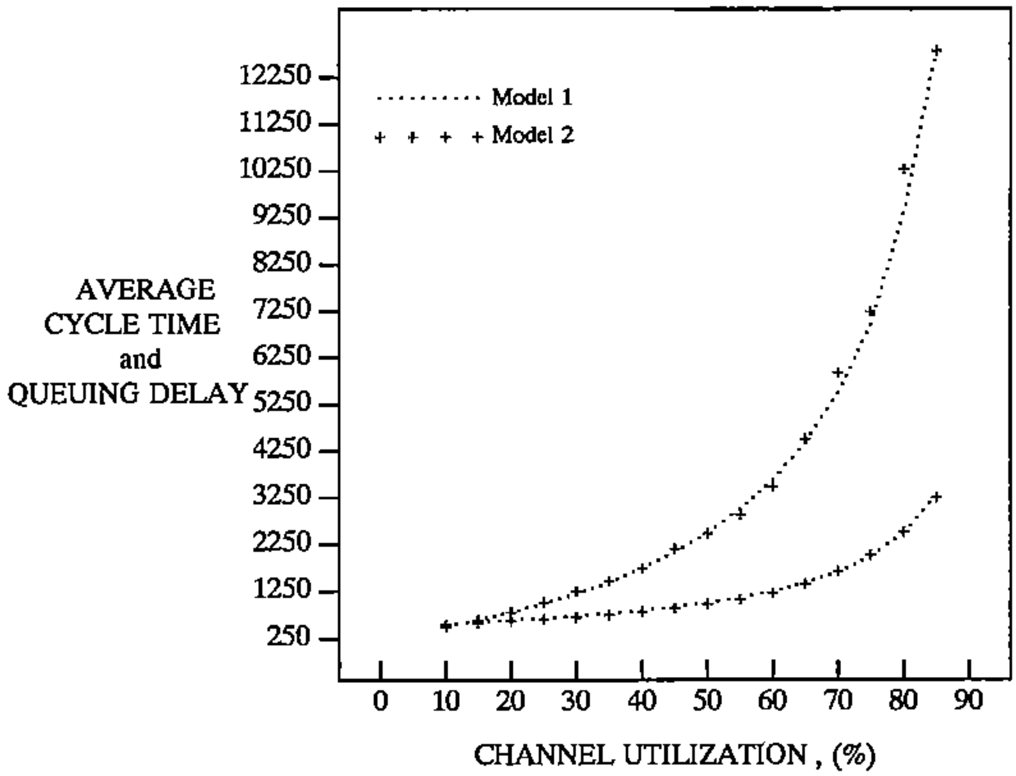

Figure 9a.

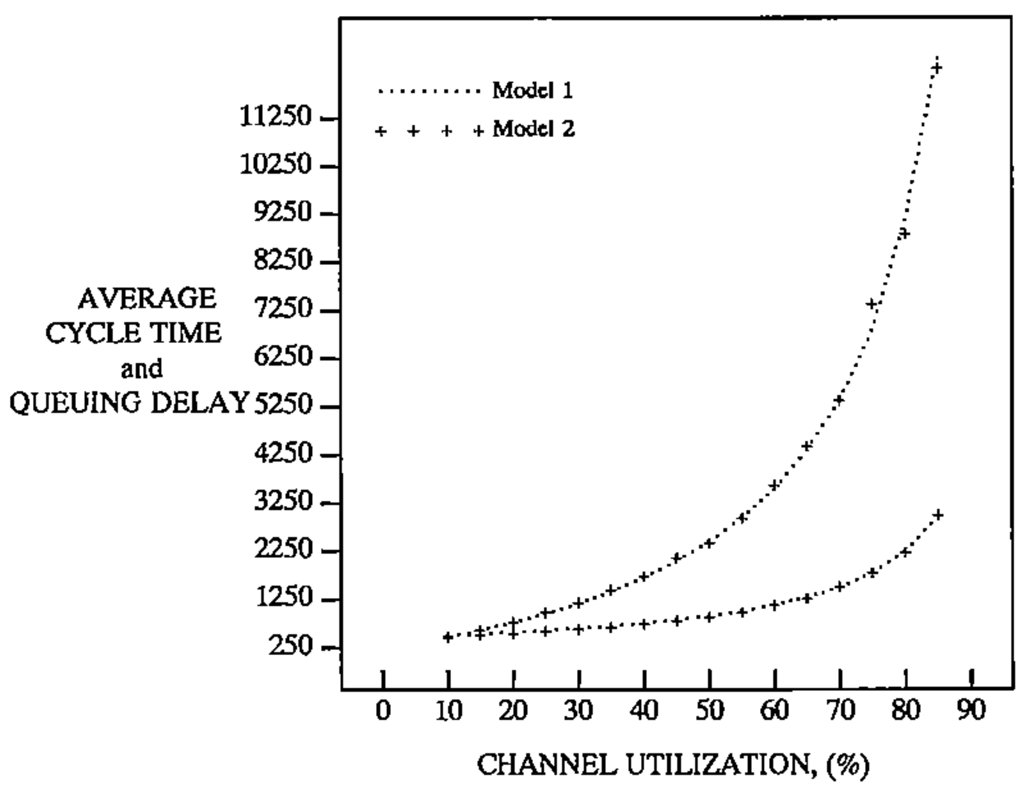

Figure 9b.

Figure 9. The average cycle time (lower curves) and average queuing delay for communication (upper curves) versus channel utilization, U_{ch}. *Transaction-oriented system of type C*, low request arrival rate (average DTU size is 2000 and average remote execution time 8000), and a communication bound system (average channel utilization twice the average server utilization). *Exhaustive service strategy at client and server queues.*

The system with request-response communication, Model 2, and the equivalent system with independent arrival rates at all nodes, Model 1. The system has a client to server ratio of: 90:10 (Figure 9a), 80:20 (Figure 9b).

The relative error in the cycle time when Model 1 is used to approximate the behavior of a request-response system, Model 2, is smaller than 2% for $U_{ch} \leq 55\%$ and is smaller than 4.0% for $U_{ch} \leq 75\%$. The corresponding upper limits for errors in queuing delays are 3.0% and 5%.

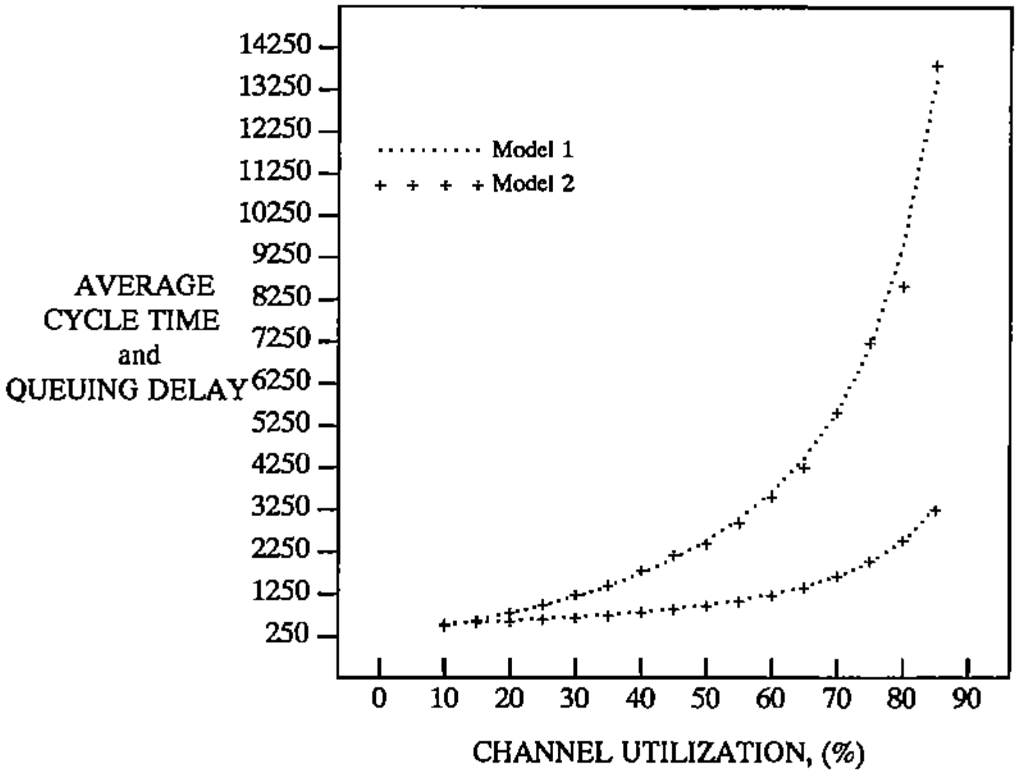

Figure 10a.

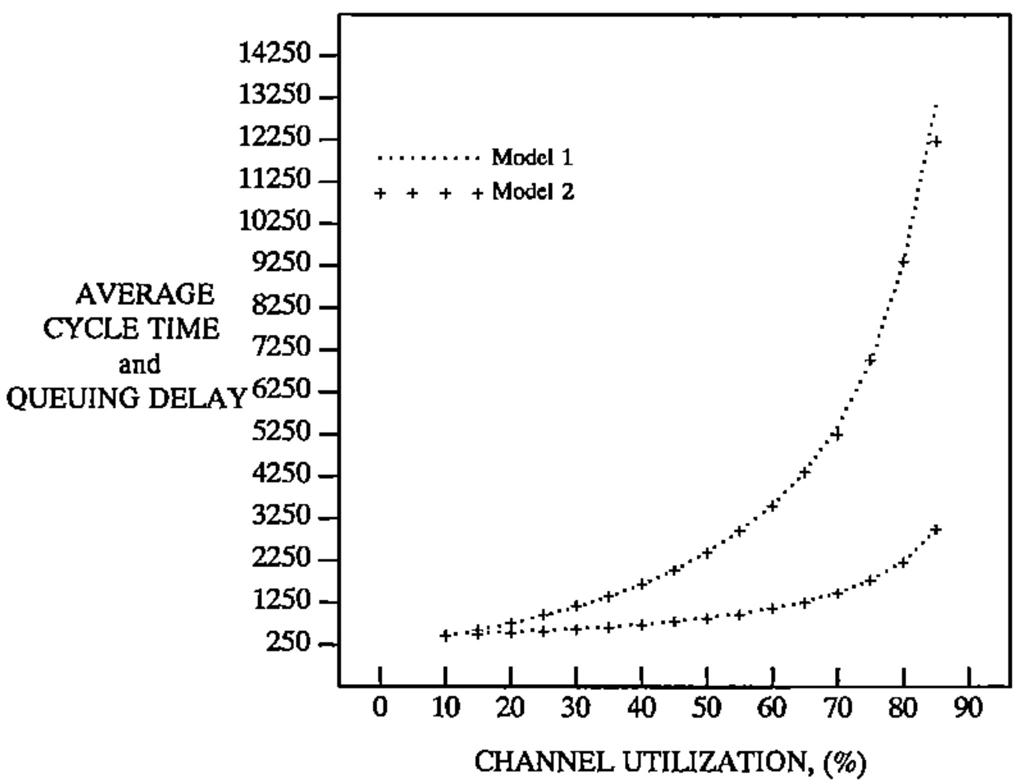

Figure 10b.

Figure 10. The average cycle time (lower curves) and average queuing delay for communication (upper curves) versus channel utilization, U_{ch}. *Transaction-oriented system of type C,* low request arrival rate (average DTU size is 2000 and average remote execution time 8000), and a communication bound system (average channel utilization twice the average server utilization). *Exhaustive service strategy at client and 1-limited at server queues.*

The system with request-response communication, Model 2, and the equivalent system with independent arrival rates at all nodes, Model 1. The system has a client to server ratio of 80:20 (Figure 10a) and 60:30 (Figure 10b).

The relative error in the cycle time when Model 1 is used to approximate the behavior of a request-response system, Model 2, is smaller than 2% for $U_{ch} \leq 55\%$ and is smaller than 3.0% for $U_{ch} \leq 75\%$. The corresponding upper limits for errors in queuing delay are 2.5% and 3%.

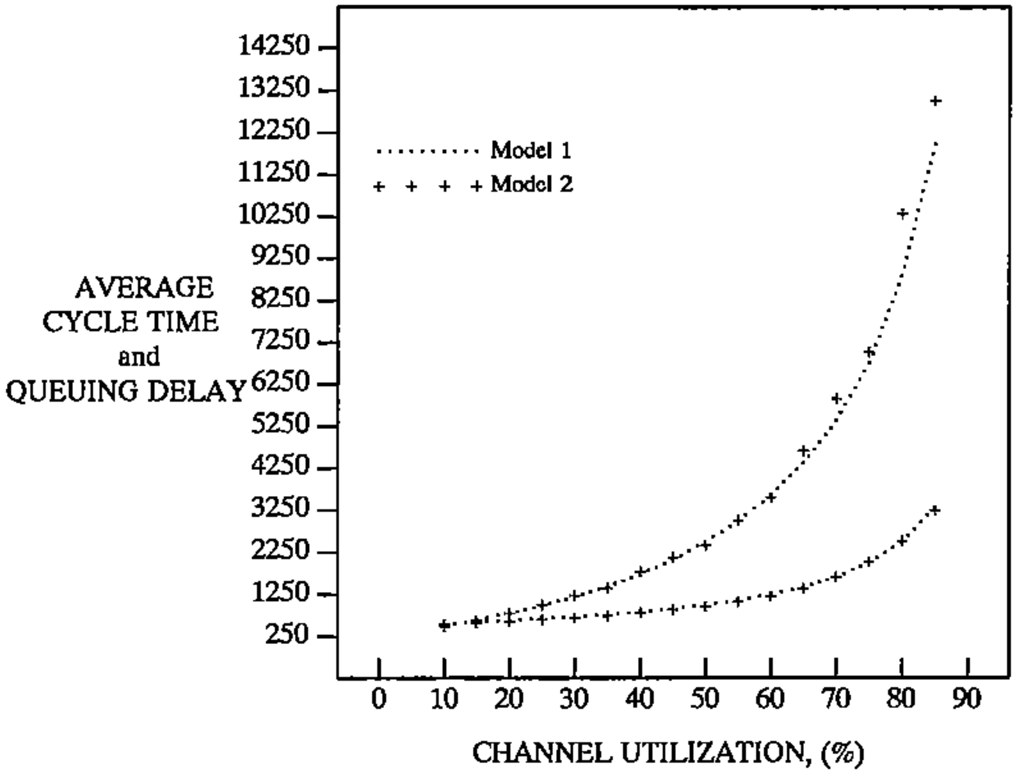

Figure 11a.

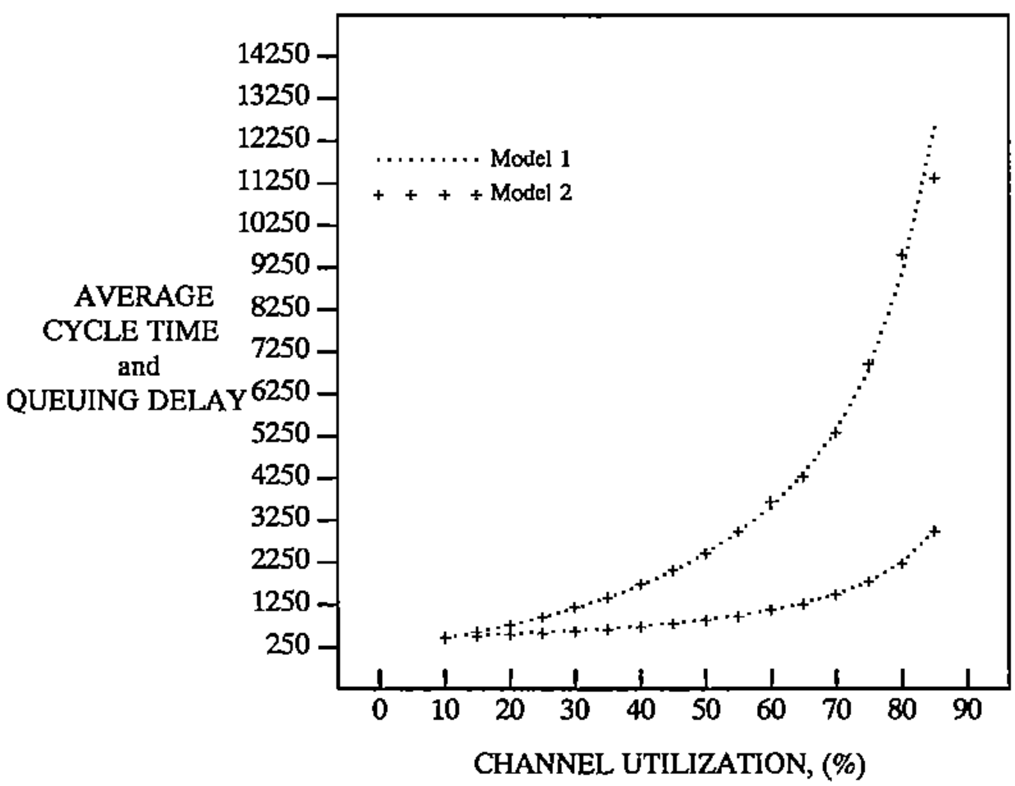

Figure 11b.

Figure 11. Average cycle time (lower curves) and average queuing delay for communication (upper curves) versus channel utilization, U_{ch}. *Transaction-oriented system of type C*, low request arrival rate (average DTU size is 2000 and average remote execution time 8000), and a communication bound system (average channel utilization twice the average server utilization). *1-limited service strategy at client and 1-limited at server queues.*

The system with request-response communication, Model 2, and the equivalent system with independent arrival rates at all nodes, Model 1. The system has a client to server ratio of: 80:20 (Figure 11a) and 60:30 (Figure 11b).

The relative error in the cycle time when Model 1 is used to approximate the behavior of a request-response system, Model 2, is smaller than 1.7% for $U_{ch} \le 0.55$ and is smaller than 2.0% for $U_{ch} \le 0.75$. The corresponding upper limits for errors in queuing delays are 1.5% and 3%.

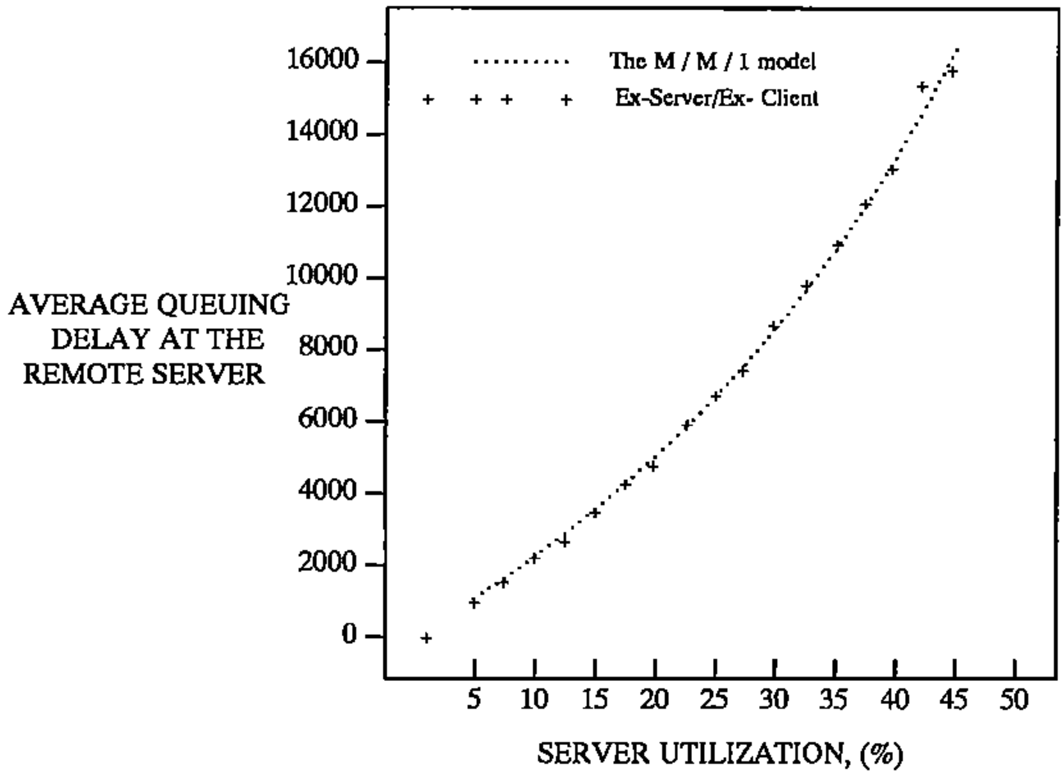

Figure 12. Average queuing delay at the remote server site function of the average server utilization. *Transaction-oriented system of type C*, low request arrival rate (average DTU size is 2000 and average remote execution time 8000), and a communication bound system (average channel utilization twice the average server utilization).

Two curves are plotted: the queuing delay as predicted by the M/M/1 system, and the queuing delay when an exhaustive service strategy is used for communication in the ring at server and at client nodes. Client to server ratio is 90:10