

Analysis of Broadcast Delivery in a Videotex System

J. W. WONG AND M. H. AMMAR

Abstract—Videotex is a system which provides users with low-cost real-time access to information. In such a system, user requests are forwarded to a service computer where the desired information is retrieved and sent back to the user. In this study, we investigate the response time behavior of a videotex system where information requested by one user is broadcast to all users. A novel queueing model for broadcast delivery is developed. Using this model, we first obtain an analytic expression for the mean response time, and then we develop an efficient algorithm for its computation. Numerical results illustrating the performance characteristics of broadcast delivery are presented.

Index Terms—Broadcast networks, computational algorithms, population size constrained networks, queueing models, response time analysis, videotex systems.

I. INTRODUCTION

Communication with pictures, as opposed to words, is not a novel concept. Ever since the first method for recording pictures was devised, engineers and scientists have been inventing procedures and systems for transmitting pictures from one place to another.

The first efforts were dedicated towards the transmission of still images over long distances, facsimile. A machine for such a purpose was designed as early as 1842. Some other early achievements included the electrical transmission of handwriting in 1886, and the sending of pictures of people along with voice signals in 1906 [1].

It is in this historical context that the development in the last decade of an interactive picture information system called *videotex* took place [2], [3]. This new system does not represent revolutionary concepts in communication. Rather, it must be seen as the result of the maturation and the coming together of various disciplines and technologies. In essence videotex provides real-time access to information, which is represented in forms that are displayable on a regular television screen or other relatively low-cost display terminals.

A typical configuration of a videotex system is shown in Fig. 1. In this system, users submit independent requests for information pages to a service computer via a communication network. When the service computer processes a request, the desired information page is retrieved from the database and broadcast to all users regardless of who requested it. This is appropriate when a broadcast medium, such as a coaxial cable or a satellite channel, is used to transmit pages to the users. This approach has the obvious advantage that the broadcast of a particular page will satisfy all outstanding requests for that page.

A queueing model of a videotex system which utilizes broadcast delivery is formally presented in Section II. In Section III analytic results for the mean response time are derived. An algorithm for the efficient computation of mean response time is presented in Section IV. Section V provides some numerical examples.

The authors are not aware of any previous studies which use queueing theoretic models to investigate the performance of broadcast delivery. Therefore, the model presented in this paper is novel, and the results are useful in characterizing the performance of videotex systems.

Manuscript received February 10, 1984; revised December 10, 1984. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

J. W. Wong is with the Department of Computer Science, University of Waterloo, Waterloo, Ont., Canada N2L 3G1.

M. H. Ammar is with the Department of Electrical Engineering, University of Waterloo, Waterloo, Ont., Canada N2L 3G1.

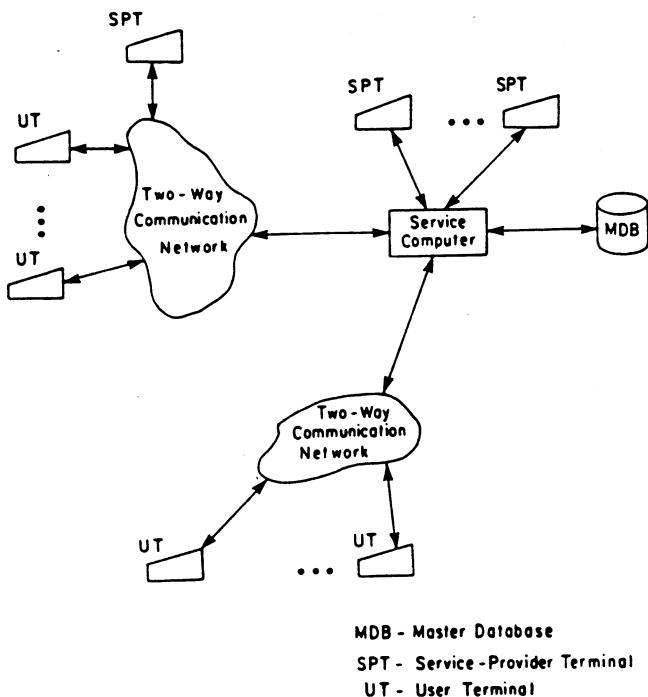


Fig. 1. A typical videotex system.

II. MODEL DESCRIPTION

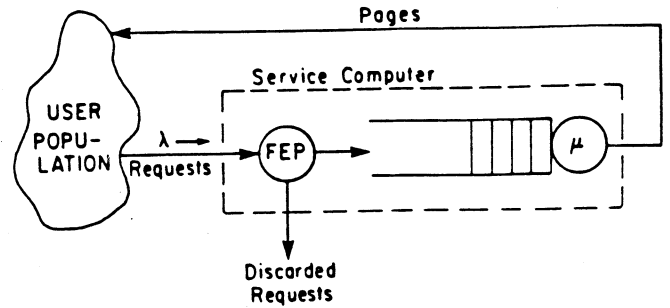
Our videotex system model is shown in Fig. 2. In this model, requests submitted by terminal users are processed by a service computer, resulting in the retrieval of the desired pages and the broadcast of these pages to all users. We assume that the delay in the broadcast network is negligible. This is reasonable when such delays are not significant compared to the queueing and processing time at the service computer, or when we are only concerned with the service computer portion of the system.

For our model, the arrival process is assumed to be Poisson with parameter λ . This assumption is appropriate for the system under consideration because the number of users is normally large. There are N possible request types, each corresponding to a particular information page. The probability that a particular request is of type i is assumed to be $q_i, i = 1, \dots, N$, where $\sum_{i=1}^N q_i = 1$. The service computer is modeled as a single-server infinite-capacity queue with an exponential service time distribution. The service rate μ is assumed to be identical for all request types, and the queueing discipline is assumed to be first-come, first-served (FCFS).

A user's request is satisfied if at any time after the request is made, a page of the same type is broadcast. Under these conditions, admitting all requests into the system may cause a broadcast that does not satisfy any outstanding requests. Such broadcast is called *superfluous*. Our goal is to reduce the number of superfluous broadcasts since they represent wastage of system resources. For this purpose, we propose to equip such a system with a front-end processor that discards a type i request if there is one already in the system (see Fig. 2). Such a request is also called *superfluous*. Note that although a superfluous request is discarded, the user that generated it will still get served. We assume the processing time at the front-end processor is negligible.

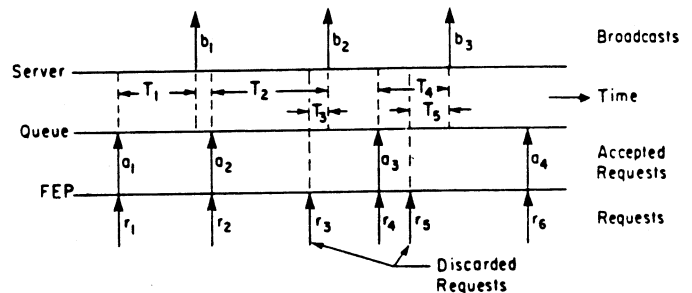
The operation of our model is illustrated in Fig. 3, where we focus on requests and page broadcasts of type i . A distinction is made in this figure between user requests and requests that actually join the queue.

The performance measure of interest is the mean response time of request type $i, i = 1, \dots, N$. This is defined to be the elapsed time from when a request for page i is made to when page i is next broadcast. Such a broadcast may be the response to an earlier request for the same page by some other user.



FEP: Front-End Processor

Fig. 2. Model for a videotex system with broadcast delivery.



r_j : j th request for page i
 a_j : j th accepted request for page i
 b_j : j th broadcast of page i
 T_j : response time of j th request

Fig. 3. Illustration of response time for the videotex system model.

III. ANALYSIS

A. Equilibrium State Probabilities

We define the state of our model to be (n_1, n_2, \dots, n_N) where $n_i, i = 1, \dots, N$, is the number of type i requests in the queueing system. Since superfluous requests are discarded, $n_i = 0$ or 1 for all i .

Let $P(n_1, \dots, n_N)$ be the probability that the system is in state (n_1, \dots, n_N) at equilibrium. Using Lam's result [4], we have

$$P(n_1, n_2, \dots, n_N) = P(0, 0, \dots, 0) \rho^n \prod_{i=1}^N q_i^{n_i} \quad (1)$$

where $n = \sum_{i=1}^N n_i$ and $\rho = \lambda / \mu$.

Let $P(n)$ be the equilibrium probability that there are n requests in the system, regardless of type. By summing over all the state probabilities in (1) where $\sum_{i=1}^N n_i = n$ we get

$$P(n) = n! \rho^n P(0) g(N, n) \quad (2)$$

where

$$g(N, n) = \sum_{\substack{n_1 + \dots + n_N = n \\ n_j \geq 1, j = 1, \dots, N}} \prod_{i=1}^N q_i^{n_i} \quad (3)$$

Since $\sum_{n=0}^{\infty} P(n) = 1$ we have

$$P(0) = \left[\sum_{n=0}^{\infty} n! \rho^n g(N, n) \right]^{-1} \quad (4)$$

B. Response Time Derivation

To derive the mean response time we define the following events:

$A_i(n) = \{n \text{ requests in system and none of type } i\}$

for $n = 0, \dots, N - 1$, and $i = 1, \dots, N$.

and

$$B_i(n, k) = \{n \text{ requests in system, } k\text{th request of type } i\}$$

for $n = 1, \dots, N, k = 1, \dots, n,$ and $i = 1, \dots, N.$

It can be shown that [5]

$$\text{Prob}[A_i(n)] = n! \rho^n P(0) g^{-i}(N, n) \quad (5)$$

and

$$\text{Prob}[B_i(n, k)] = (n - 1)! \rho^n P(0) q_i g^{-i}(N, n - 1) \quad (6)$$

where

$$g^{-i}(N, n) = \sum_{\substack{n_1 + \dots + n_N = n \\ n_i = 0, n_j \geq 1, j \neq i}} \prod_{l=1}^N q_l^{n_l} \quad (7)$$

If at the moment of the arrival of a type i request, event $A_i(n)$ is satisfied, then the request is accepted, and its mean response time will be $(n + 1)/\mu$. If, on the other hand, event $B_i(n, k)$ is satisfied, the request is discarded, and its mean response time will be k/μ .

The mean response time for a page i request can now be found by summing over all the possible system states. Using (5) and (6) we get

$$S_i = \frac{P(0)}{\mu} \sum_{n=0}^N (n + 1)! \rho^n \left[g^{-i}(N, n) + \frac{q_i}{2} g^{-i}(N, n - 1) \right] \quad (8)$$

where $P(0)$ is given by (4) and $g^{-i}(N, -1) = g^{-i}(N, N) = 0$.

The mean response time over all page types is

$$S = \sum_{i=1}^N q_i S_i$$

C. Limiting Behavior at Heavy Load

In this section, we investigate the limit of S_i as ρ approaches infinity. This corresponds to the case where the request arrival rate far exceeds the page retrieval rate. Note that the maximum queue length is N because superfluous requests are discarded. The system is, therefore, stable, provided that N is finite.

It can be shown using L'Hopital's rule on (8) that

$$\lim_{\rho \rightarrow \infty} S_i = \frac{N + 1}{2\mu} \quad \text{for all } i = 1, \dots, N. \quad (9)$$

We interpret the above limit by observing that as ρ increases, the probability that there are N requests in the system approaches 1. All these requests have to be distinct. Thus, the probability that, upon the arrival of a request of type i , the same type request is in the k th position approaches $1/N$, in which case its mean response time is k/μ . Hence,

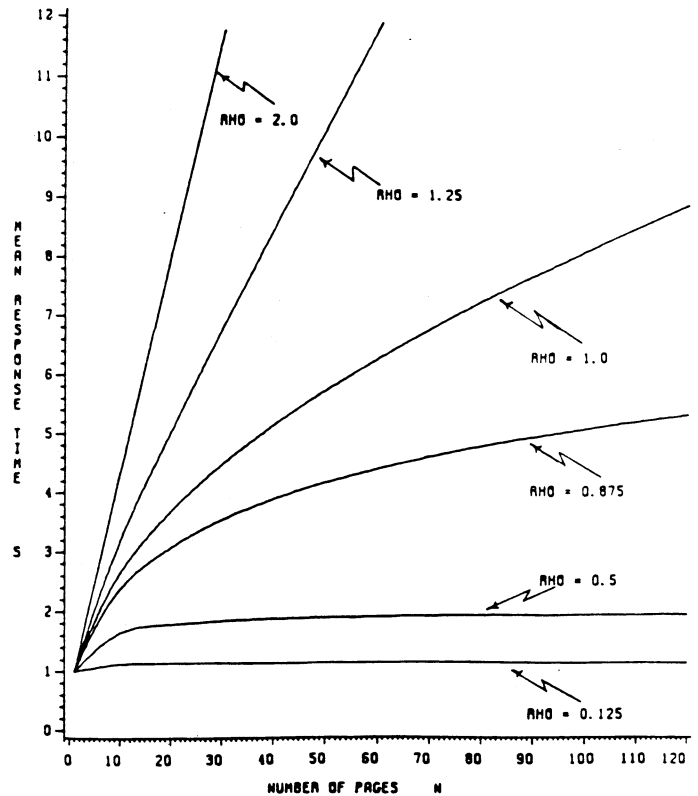
$$\lim_{\rho \rightarrow \infty} S_i = \sum_{k=1}^N \frac{k}{\mu} \left(\frac{1}{N} \right) = \frac{N + 1}{2\mu}$$

IV. COMPUTATION OF MEAN RESPONSE TIME

We now discuss techniques that allow for the efficient numerical computation of S_i . Clearly, the major hurdle in evaluating (8) is the computation of the functions $g(N, n)$ and $g^{-i}(N, n)$ for an arbitrary request probability vector $\{q_1, \dots, q_N\}$. The computational requirement of conventional algorithms is on the order of 2^N [7], [8]. This may present some difficulties for large N . In this section we first present some useful recursive relationships, and then illustrate how they can be used to compute $g(N, n)$ and $g^{-i}(N, n)$ efficiently.

A. Recursive Relationships

Using a technique first developed by Buzen [6], it can be shown that the functions $g(N, n)$ and $g^{-N}(N, n)$ satisfy the following relationships [5].



N Equally Likely Pages, $\mu = 1.0$

Fig. 4. Mean response time versus number of pages.

Relationship 1:

$$g(N, n) = g(N - 1, n) + q_N g(N - 1, n - 1). \quad (10)$$

Relationship 2:

$$g^{-N}(N, n) = g(N - 1, n). \quad (11)$$

B. Computational Algorithm

Using (10), we develop the following algorithm to calculate $g(N, n), n = 0, \dots, N$, for an arbitrary request probability vector $\{q_1, \dots, q_N\}$.

Declare $G(0 \rightarrow N)$ Array of Reals

Initialize:

$$G(0) \leftarrow 1$$

$$G(1) \leftarrow q_1$$

$$G(2 \rightarrow N) \leftarrow 0$$

For $i = 2$ to N

For $j = i$ to 1 by -1

$$G(j) \leftarrow G(j) + q_i * G(j - 1)$$

End

End

The required functions are contained in the array G at the end of the algorithm, i.e., $g(N, n) = G(n)$.

The above algorithm can also be used to calculate $g^{-N}(N, n)$ since by Relationship 2 it is equal to $g(N - 1, n)$. To calculate $g^{-i}(N, n)$ for an arbitrary i , we use the algorithm with the rearranged request probability vector $\{q_1, \dots, q_{i-1}, q_N, q_{i+1}, \dots, q_i\}$ as input.

Our algorithm requires on the order of N^2 steps to calculate $g(N, n)$. However, to calculate $g^{-i}(N, n)$ for all i , as described above, the algorithm will have to be applied N times. Thus, a complete solution will require on the order of $N^2 + N^3$ steps. The storage requirement is N elements.

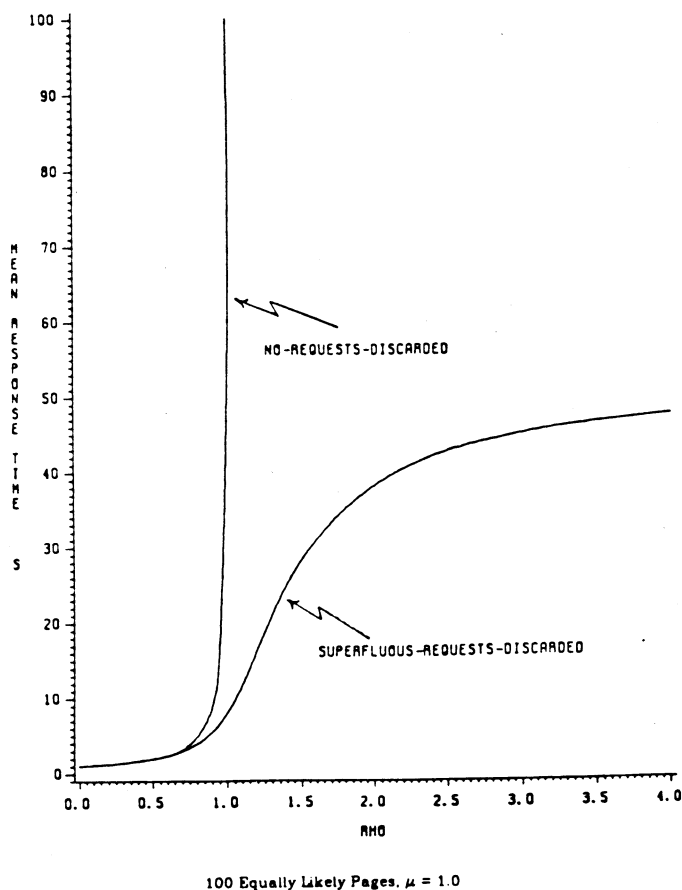


Fig. 5. A comparison of systems with and without superfluous request discarding.

V. NUMERICAL EXAMPLES AND DISCUSSION

The analytic results in Section III are for a general request probability distribution. We focus here, for simplicity, on the case where each page is equally likely to be requested, i.e., $q_i = 1/N$ for all $i = 1, \dots, N$. Note that, for this case, $S = S_i$ for any i .

Fig. 4 shows how S varies as a function of N for various values of ρ . We observe the following two properties.

1) For $\rho \geq 1$, the mean response time is a linear function of N as $\rho \rightarrow \infty$. (See Section III-C.)

2) For $\rho < 1$, as N increases, the mean response time approaches a constant. For the case $q_i = 1/N$, it can be shown that (8) yields a mean response time given by the $M/M/1$ result, i.e., $1/(\mu(1 - \rho))$ when N is large. This can be explained by the fact that when q_i is small for all i , the probability that an arriving request is superfluous approaches zero. Note that if q_i does not decrease with N for some i , then the mean response time at large N is different from that of the $M/M/1$ model.

Fig. 5 shows the advantage of discarding superfluous requests. In [5] analytic results were obtained for the case where superfluous requests are not discarded. These results can be summarized as follows:

$$S_i = \frac{1}{\mu(1 - \rho(1 - q_i))} \quad (12)$$

and

$$S = \sum_{i=1}^N \frac{q_i}{\mu(1 - \rho(1 - q_i))}. \quad (13)$$

We observe that discarding superfluous requests leads to better mean response time. At heavy load, the queue becomes unbounded if superfluous requests are not discarded. The limit of the mean response time in (13) as $\rho \rightarrow 1$ (condition of a saturated system when superfluous requests are not discarded) is N/μ . This is approximately twice the heavy load limit in (9) for the case where superfluous requests are discarded.

REFERENCES

- [1] W. H. Ninke, "Interactive picture information systems—Where from? Where to?" *IEEE J. Select. Areas Commun.*, vol. SAC-1, Feb. 1983.
- [2] J. Martin, *Viewdata and the Information Society*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
- [3] J. Gecsei, *The Architecture of Videotex Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
- [4] S. S. Lam, "Queueing networks with population size constraints," *IBM J. Res. Develop.*, vol. 21, no. 7, July 1977.
- [5] M. H. Ammar and J. W. Wong, "Analysis of broadcast delivery in videotex systems," *Comput. Commun. Networks Group, Univ. Waterloo, Waterloo, Ont., Canada, Rep. E-118, Jan. 1984*.
- [6] J. P. Buzen, "Computational algorithms for closed queueing networks with exponential servers," *Commun. ACM*, vol. 16, no. 9, Sept. 1973.
- [7] M. Reiser and H. Kobayashi, "Queueing networks with multiple closed chains: Theory and computational algorithms," *IBM J. Res. Develop.*, vol. 19, no. 3, May 1975.
- [8] S. Bruell and G. Balbo, *Computational Algorithms for Closed Queueing Networks*. Amsterdam, The Netherlands: North-Holland, 1980.