Selective Broadcast Data Distribution Systems

K. H. Yeung, *Member, IEEE*, and
T.S. Yum, *Senior Member, IEEE*

Abstract—This paper describes a two tier architecture for high speed data distribution. The architecture consists of a database interface network which distributes information from a central database to a number of servers, and a user interface network which distributes information from the servers to the user terminals. The database interface network uses the *Selective Broadcast* technique to distribute data on a high speed channel. Analytical results and design examples showed that Selective Broadcast technique can provide an order of magnitude smaller response time under normal traffic conditions when compared to the nonselective broadcast technique such as the Datacycle™ system.

Index Terms—Data distributions, selective broadcast, high speed networks, information delivery systems, information retrieval systems.

1 Introduction

Recent advances in computer and communication technologies have led to the development of information delivery systems that provide users with real time access to a broad spectrum of information. Examples of such information delivery systems are Videotex systems, multimedia information systems, digital news systems, and systems for medical imaging applications. Conventional centralized information delivery systems are based on the central server model in which a central service computer replies to each user request in an individual response manner. The main drawback of this approach is the rapid increase of response time as the system load approaches the server's capacity. This situation can be improved with the introduction of the broadcast delivery and mixed delivery techniques [1], [2], [3], [4]. In [4], it was shown that the response time using these techniques is significantly smaller than those based on the individual response model. However, the fact that a central server still remains in broadcast delivery models means that the limited power of the server is still the potential bottleneck of the overall information flow.

The basic configuration of systems based on the central server model consists of a database, a service computer, a communication network, and user terminals. Information is organized into units called *pages*, and stored in a database. Users make requests and receive the requested pages through their terminals. The service computer retrieves the requested pages and transmits them to the user terminals via a communication network. Instead of considering database systems where there are many record updates, we consider only the read-oriented information delivery systems in this paper.

2 The Distributed Architecture

The systems described above have two potential bottlenecks: throughput of the communication network and I/O speed and processing power of the service computer. The relative significance of these two potential bottlenecks is application specific and depends on many factors. For future information systems with extremely large databases, the main bottlenecks will very likely be the service computer.

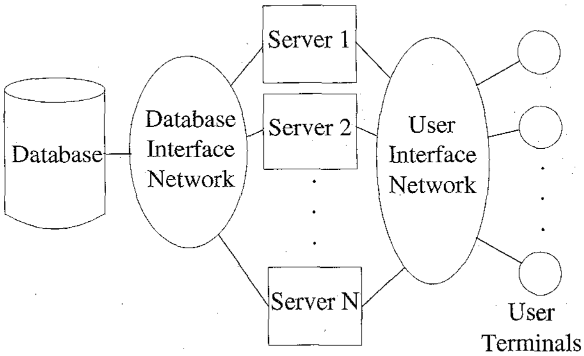

Fig. 1. A modified configuration of information delivery system.

A multiple server architecture (Fig. 1) is studied in this paper. Here, the multiple servers working in parallel has the advantages of faster response due to distributed processing, and modular growth in the number of servers. The database interface network and the user interface network parts are detailed in the following subsections.

2.1 Database Interface Network

The function of the database interface network is to distribute information from the central database machine to the servers. One way to do this is to use the Datacycle™ technique [5] whereby the entire database is pumped out from the database machine and distributed to the servers through a high speed link and the servers filter out the information required by the users. This technique has two advantages. First, it is relatively easy to optimize the I/O performance for sequential accesses. Second, the load on the database machine is independent of the volume of the traffic generated by the users. A performance analysis of Datacycle™ and a new concurrency control scheme for such use is given in [6].

A new technique called the *Selective Broadcast* technique is proposed in this paper for use in the database interface network to minimize the delay due to long cycle time of data. The data being broadcast is organized into units called *blocks* with each block consisting of multiple pages. Let there be a total of B blocks. The technique is based on the observation that in most cases only a small percentage of the data being pumped out is actually required; and so if a block is broadcast only when a confirmation for that block is received by the database machine, the data cycle time can be shortened to a small fraction of the original. The confirmation is done via the *Confirmation Ring* (Fig. 2) which connects the database machine and all the servers. Periodically the database machine sends out a B bits frame to the ring. These B bits serve as a bit map of the B blocks of data. The frame circulates through all the N servers with a one bit delay on each. When the frame returns to the database machine, a new frame is sent for the next round of confirmation. A new frame generated by the database machine has a content of all zeros. If a server wants to confirm the *i*th block (due to a request from a user it is serving, say), it simply write a 1 to the *i*th bit of the frame. Otherwise the server simply passes the frame to the next server without modification. The content of the frame, therefore, reflects the specific blocks being confirmed.

At the database machine, the returned frame is copied into a B-bit *Block-Confirm (BC) register*. The block in transmission is marked by a pointer on the BC register. Each time the database machine is ready to send a block, it advances the pointer to the next nonzero bit and transmits the corresponding confirmed block. When the pointer reaches the end of the register, it cycles back to the beginning.

- *K.H. Yeung is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. E-mail: eeayeung@cityu.edu.hk.*
- *T.S. Yum is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. E-mail: yum@ie.cuhk.edu.hk.*

Manuscript received June 20, 1994; revised March 1996.
For information on obtaining reprints of this article, please send e-mail to: transcom@computer.org, and reference IEEECS Log Number C96298.

0018-9340/97$10.00 ©1997 IEEE

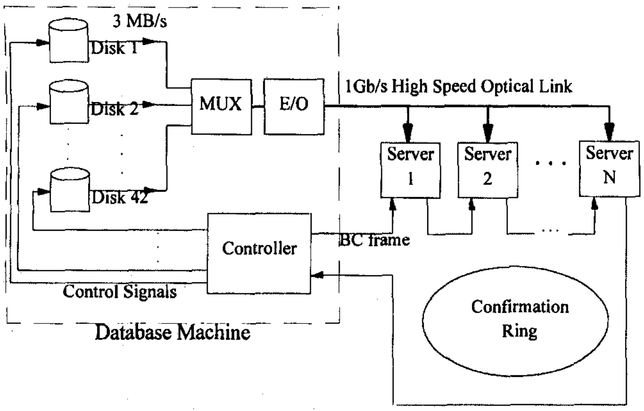

Fig. 2. The database interface network and the central database machine.

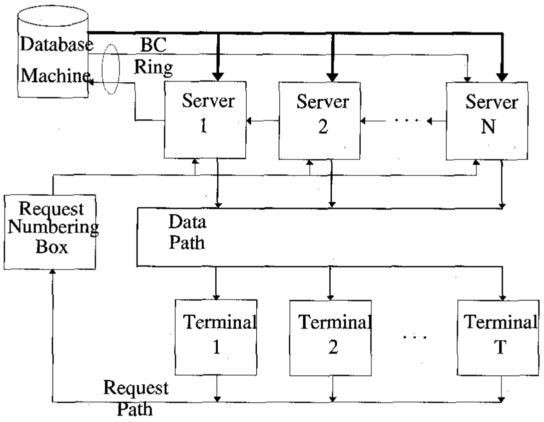

Fig. 3. The distributed data distribution system.

Note that the BC register contains the confirmation status of the entire database, so advance functions such as prefetching of data blocks from the database can also be incorporated. The analysis of such mechanisms, however, is beyond the scope of this paper.

A large number of RAID architectures can be used to implement the database machine. A very simple example is shown in Fig. 2 where 42 inexpensive disks of data rate 3 Mbyte/s are multiplexed onto a 1 Gb/s high-speed optical link.

2.2 User Interface Network

The function of the user interface network is to exchange data between the servers and the user terminals. We assume the user interface network is an interconnection of many LANs and MANs serving many users. A large variety of LANs and MANs can serve this purpose. Discussion of their relative merit, however, is beyond the scope of this paper.

2.3 Operation of the System

Fig. 3 shows a block diagram of the distributed data distribution system. When a page request is issued by a user, it is sent to the request numbering box (RNB) through the request path. The RNB has the simple function of distributing the requests according to the processing rates of the servers. A detailed study of how to match the bifurcated traffic rates to the processing rates of the servers is beyond the scope of this paper. As no instantaneous queue length information of the servers is available to the RNB, the load balancing policy used can only be of the static type. For the most common case where all servers are identical, the RNB has the simple function of distributing the requests to the servers in a round-robin fashion. A server performs two types of operations, as follows. At the user interface side, when a server receives a page request it uses its internal page directory to identify the particular block needed and makes a confirmation on that block. On the database interface side, a server filters out all its required blocks from the database interface network, extracts the requested pages from these blocks, and sends them to the user terminals through the data path. The page directory is stored as a directory block in the database. When a server is powered up the directory block is first retrieved from the database and loaded into the server.

3 Mean Block Acquisition Delay for Uniform Request Distribution[1]

3.1 Upper Bound Derivation

In the following analysis, time is measured in *slots*, one slot being the time required to broadcast one *block* of data on the database interface network. The arrival of requests is assumed to be a Poisson process of rate λ per slot. In this section, we study the case where the request distribution is uniform, or the probability that a certain block is requested is identical for all blocks. The analysis for general request distributions will be given in the next section.

Let $\gamma(n)$ be the number of confirmed blocks waiting in the transmission queue at the database machine at slot n. $\gamma(n)$, therefore, does not include the one in transmission. We shall, for convenience, call it the backlog size. Since the backlog size at slot $n + 1$ depends only on $\gamma(n)$ and the number of requests in slot n, the evolution of $\gamma(n)$ is a discrete time Markov chain.

Let random variable A denote the number of requests per slot and random variable K denote the total number of blocks for which the A requests are located. As the requests are assumed to be randomly located in the database, the probability that a requests fall in k blocks is given in [8] as

$$P[K=k|A=a]=B^{-a}\binom{B}{k}\sum_{y=0}^{k}(-1)^{y}\binom{k}{y}(k-y)^{a}. \qquad (1)$$

Removing the conditioning on A, we obtain

$$P[K=k]=\sum_{a=0}^{\infty}P[K=k|A=a]\frac{\lambda^{a}e^{-\lambda}}{a!} \qquad k=0,1,2,\ldots,B. \qquad (2)$$

Next, let random variable M_i be the number of arrivals whose requested blocks need to be confirmed when the backlog is i. We shall, for convenience, call these arrivals the *new* customers and those arrivals that do not generate confirmations the *subsequent* customers. If $M_i = m$, the backlog at the next slot will be $i - 1 + m$. Given that the backlog is i and k blocks are requested at the current slot, the probability that m out of these k blocks are to be confirmed is

$$\begin{aligned} &P[M_i=m|K=k] \\ &=\frac{\begin{pmatrix}\text{number of ways to choose the } m \text{ request}\\ \text{blocks from the } (\mathrm{B}-i) \text{ unconfirmed blocks}\end{pmatrix}\times\begin{pmatrix}\text{number of ways to choose the remaining } k-m\\ \text{request blocks from the } i \text{ confirmed blocks}\end{pmatrix}}{\begin{pmatrix}\text{number of ways to choose } k\\ \text{request blocks from B blocks}\end{pmatrix}} \qquad (3)\\ &=\frac{\binom{\mathrm{B}-i}{m}\binom{i}{k-m}}{\binom{\mathrm{B}}{k}} \qquad m=0,1,2,\ldots,k \end{aligned}$$

[1] This section was reported in [7].

Removing the conditioning on K, we obtain the distribution of *new* customers in a slot as

$$P[M_i = m] = \sum_{k=0}^{B} P[M_i = m | K = k] P[K = k] \tag{4}$$

At steady state, the transition probabilities are given by

$$h_{ij} \triangleq P[\gamma(n+1) = j | \gamma(n) = i] = \begin{cases} 0 & for\, j < i-1 \\ P[M_i = j-i+1] & for\, j \ge i-1 \end{cases} \tag{5}$$

Having obtained the transition probabilities, the equilibrium distribution of the backlog size, denoted as $\{\pi_0, \pi_1, \ldots, \pi_B\}$ can be computed in the usual way. The average waiting time of the new customers, denoted by E[W_{new}], is given by the Little's formula as

$$E[W_{new}] = \frac{E[\gamma]}{E[M]} = \frac{\sum_{i=1}^{B} i\pi_i}{\sum_{m=0}^{B} m \sum_{i=1}^{B} \pi_i P[M_i = m]} \tag{6}$$

Note that W_{new} is the waiting time experienced by the new customers. The subsequent customers will experience a smaller waiting time as the block request was already placed by the new customers. The expected waiting time of all customers E[W] can be computed as follow. Consider the arrival of a new customer and its subsequent departure from queue after a waiting time W_{new} slots. During its stay in the queue, subsequent customers $S_1, S_2, \ldots, S_j$ for the same block will arrive. The arrival rate is λ/B per slot. Let us condition on the event $W_{new} = t$. Since the arrival of the subsequent customers is a Poisson process, their arrival times are uniformly distributed in interval [0, t] and their average delay is just t/2. Let there be j such arrivals. Then the average waiting time of these j + 1 customers is

$$E[W|j,t] = \frac{t + j\frac{t}{2}}{j+1} \tag{7}$$

Removing the conditioning on j, we have

$$E[W|t] = \sum_{j=0}^{\infty} \left(\frac{t + j\frac{t}{2}}{j+1} \right) \left(\frac{e^{-\lambda t/B} (\lambda t / B)^j}{j!} \right) = \frac{t}{2} + \frac{B\left(1 - e^{-\lambda t/B}\right)}{2\lambda} \tag{8}$$

The evaluation of E[W] requires the distribution of W_{new} which is not available. But the use of Jensen's inequality [9] allows us to obtain an upper bound on E[W]. It is easy to show that (8) is a convex $\cap$ function of t and therefore the inequality gives

$$E[W] \le \frac{1}{2\lambda} \left\{ \lambda E[W_{new}] + B\left(1 - e^{-\lambda E[W_{new}]/B}\right) \right\} \tag{9}$$

A plot of (9) shows that the curve is fairly flat for typical values of B and λ. We would therefore expect the bound to be very tight. This is confirmed by the numerical results presented below. Finally the mean block acquisition delay E[T] is simply E[T] = E[W] + 1.

After acquiring the blocks of data, the servers need to process them and deliver them to the users. The processing delay is usually a small fixed quantity independent of the system traffic. The delivery delay depends on the actual delivery network (usually a LAN) and its load. The mean response time for systems using Selective Broadcast technique is the sum of the three delays. The focus of the present study will only be on the mean block acquisition delay E[T].

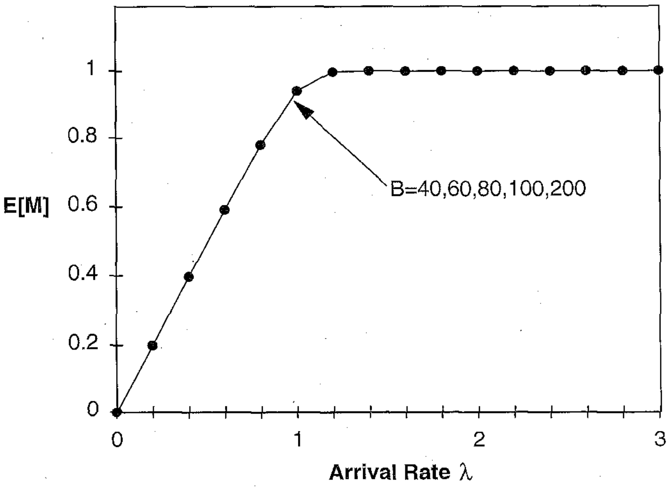

Fig. 4. The expected number of new arrivals against arrival rate.

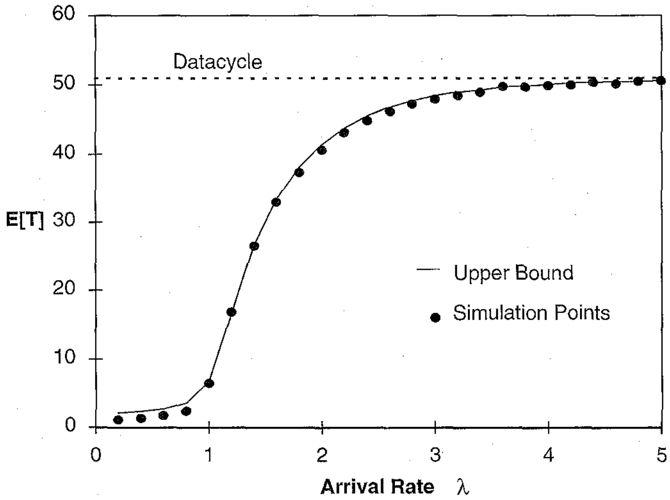

Fig. 5. The upper bound on the mean block acquisition delay E[T] for Selective Broadcast technique when B = 100.

3.2 Numerical Examples and Simulation Results

Numerical examples are given here to compare the average block acquisition delay for the Selective Broadcast technique and that for the Datacycle™. For Datacycle™, blocks are broadcast from the database machine sequentially with each block appearing exactly once in each cycle. The mean block acquisition delay for Datacycle™ is simply (B/2) + 1 slots and is independent of the request traffic.

Fig. 4 shows the expected number of new arrivals per slot (i.e., E[M]) against the arrival rate for the five cases: B = 40, 60, 80, 100, and 200. We observe that E[M] is practically independent of B. It grows linearly between $0 \le \lambda \le 1$ and saturates at 1 when $\lambda > 1$. This is expected from our data (not shown) that the mean backlog size is a very small fraction of B in the range $0 \le \lambda \le 1$ and so almost all arrivals are "new" customers.

Fig. 5 compares the upper bound of the mean block acquisition delay E[T] with the simulation results for Selective Broadcast technique when B = 100. The 95% confidence intervals are all smaller than the size of the symbol "•" shown in the figure. The figure shows that the upper bound on E[T] is in fact very tight. Another observation is that the Selective Broadcast technique provides at least an order of magnitude smaller delay than that of the Datacycle™ (which has a constant delay of 51) at the traffic level of $\lambda \le 0.8$. Selective Broadcast technique also provides uniformly lower delay than the Datacycle™ under all traffic conditions. The results for B = 200 is similar and therefore not shown.

4 Mean Block Acquisition Delay for General Request Distributions

In this section, we first present an approximate derivation of E[T] for general request distributions. We then show that the approximation is very accurate by comparing it with simulation results.

4.1 Approximate Analysis

Let λ_i be the Poisson arrival rate of the requests for block i, and $\sum_{i=1}^{B} \lambda_i = \lambda$. We define a *cycle* as the period from the instance that a certain block has a chance to transmit until the instance it has the next chance to transmit. Obviously, blocks are not transmitted if they are not confirmed. Let random variable N be the cycle length in blocks and random variables n_i be the number of times block i appears in a cycle. Obviously, $N \geq 1$ for a cycle to exist and $n_i \in \{0, 1\}$. We thus have $N = n_1 + n_2 + \ldots + n_B$ under the condition that not all n_is are zero. A bound on N without the attached condition is therefore

$$N \leq 1 + n_1 + n_2 + \ldots + n_B. \tag{10}$$

As the n_is are all nonnegative random variables, we can take expectation to obtain

$$E[N] \leq 1 + \sum_{i=1}^{B} P[n_i = 1] \tag{11}$$

where P[n_i = 1] is the probability that block i is confirmed (i.e., will appear in a cycle) and is given by

$$\begin{aligned} P[n_i = 1] &= P[\text{at least one block } i \text{ arrival in a cycle}] \\ &= 1 - e^{-\lambda_i N}. \end{aligned} \tag{12}$$

Since the distribution of N is not available, the best we can do is to use Jensen's inequality [9] to obtain an upper bound on P[n_i = 1]. It is easy to show that (12) is a convex $\cap$ function of N and therefore P[n_i = 1] is bounded by

$$P[n_i = 1] \leq 1 - e^{-\lambda_i E[N]} \tag{13}$$

Substitute into (11), we get

$$E[N] \leq (B + 1) - \sum_{i=1}^{B} e^{-\lambda_i E[N]} \tag{14}$$

Let g(E[N]) denote the R.H.S. of (14).

LEMMA 1. *There is a unique solution denoted as x for* E[N] = *g*(E[N]) *in the interval* (0, *B* + 1).

PROOF. Fig. 6 shows E[N] and g(E[N]), and we observe that

1) E[N] < g(E[N]) at E[N] = 0.
2) g(E[N]) is a strictly increasing function of E[N].
3) E[N] > g(E[N]) at E[N] = B + 1.

Therefore, there exists one and only one solution in (0, B + 1). □

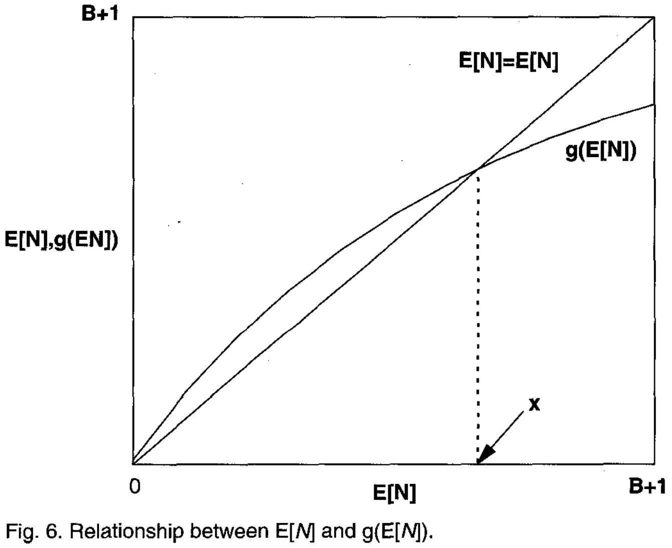

Fig. 6. Relationship between E[N] and g(E[N]).

THEOREM 1. *The expected number of confirmed blocks per cycle is bounded by x.*

PROOF. Referring to Fig. 6, we observe that E[N] ≤ g(E[N]) for E[N] ≤ x and E[N] > g(E[N]) for E[N] > x. Therefore, the inequality given by (14) is satisfied only when E[N] ≤ x. Substituting x into (14), we obtain

$$E[N] \leq (B + 1) - \sum_{i=1}^{B} e^{-\lambda_i x} = x \tag{15}$$

Having obtained an upper bound on E[N], the mean block acquisition delay E[T] can be approximated by:

$$\begin{aligned} E[T] &= \text{average waiting time} + \text{block transmission time} \\ &= \frac{x}{2} + 1 \end{aligned} \tag{16}$$

□

4.2 Numerical Examples and Simulation Results

To verify the analytical results obtained above, we perform simulations on three typical request distributions: uniform distribution, Zipf's distribution, and geometrical distribution. The Zipf's distribution [10], stipulates that block i is requested with probability c/i where c is the normalization constant. For geometrical distribution, the request probability for block i is equal to $c\rho^i$ where c is the normalization constant and ρ is the skewing factor. We observe from the results shown in Fig. 7 that for all three request distributions, the approximation on E[T] is very accurate compared to the simulation results. The results for the Zipf's and geometrical distributions show that Selective Broadcast technique performs better for more skewed distributions. This is expected because more requests are identifying on a smaller set of popular blocks with skewed distributions.

5 Optimal Choice of Block Sizes

To determine the optimal block size, we redefine a slot to be the time required to broadcast one *page* (instead of one block) of data on the database interface network. Let b pages be grouped into a block and let the database has a total size of L pages. The number

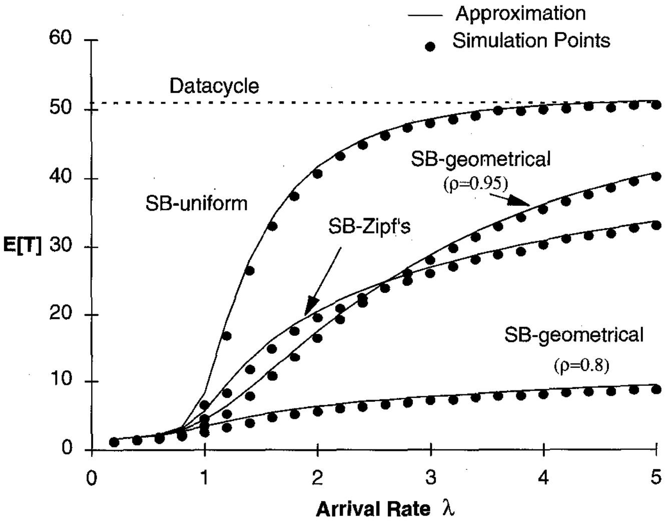

Fig. 7. Approximation on the mean block acquisition delay E[T] for Selective Broadcast technique when B = 100.

of blocks B is then equal to $\lceil L/b \rceil$. The time for the database machine to locate a certain block on the disk is assumed to be a constant of d slots. With that the transmission time for a block is $b + d$ slots. One recent study on the HP-UX (Unix) computer systems shows that half of the I/O operations have nearly constant mean I/O time if cached disks are used [11]. Since the disk in a Selective Broadcast system operates sequentially with skips, I/O requests will therefore frequently hit the cache memory of the disk. The disk delay can thus be assumed to be closed to a constant for most I/O requests. With that, the approximate analysis given in the previous section applies directly by equating one "block" time unit to $b + d$ "page" time units.

Fig. 8 shows how the block size b affects E[T] for Zipf's distribution when $d = 1$. At $\lambda = 1$ and $\lambda = 10$, b is optimal for a wide range between five and 100. When b is smaller than five, the disk delay will dominate the transmission overheads. On the other hand, when b is larger than 100, the time spent in broadcasting the nonrequested pages becomes the major transmission overheads. At lower traffic levels, such as $\lambda = 0.1$, the system always favors smaller b because the number of requested pages per confirmed block is close to one. The time wasted in broadcasting other nonrequested pages in a confirmed block will be minimized if a smaller b is chosen. For uniform distribution, our data (not shown) shows that similar conclusions can be drawn.

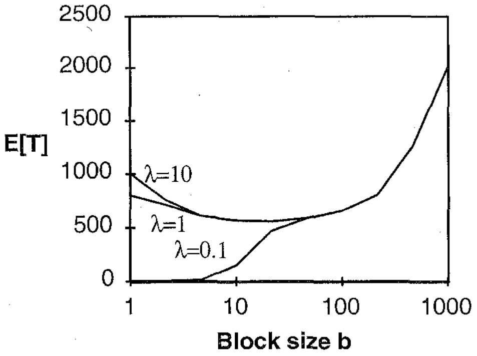

Fig. 8. The effect on changing the block size b for Zipf's distribution. The curves show the approximation on the mean block acquisition delay at different arrival rates.

6 Conclusions

In this paper, a new architecture for very high speed data distribution is proposed. The architecture consists of two separate networks: the database interface network and the user interface network. A new technique called the Selective Broadcast technique is used in the database interface network for high speed data distribution. Numerical examples show that the Selective Broadcast technique can give much smaller block acquisition delay than the Datacycle™ technique under nonoverload conditions.

Acknowledgment

The authors would like to thank the reviewers for their many helpful suggestions.

References

[1] J.W. Wong and M.H. Ammar, "Response Time Performance of Videotex Systems," *J. Selected Areas in Comm.*, vol. 4, no. 7, pp. 174-180, Oct. 1986.

[2] M.H. Ammar, "Response Time in a Teletext System: An Individual User's Perspective," *IEEE Trans. Comm.*, vol. 35, no. 11, pp. 1,159-1,170, Nov. 1987.

[3] D.K. Gifford, J.M. Lucassen, and S.T. Berlin, "The Application of Digital Broadcast Communication to Large Scale Information Systems," *J. Selected Areas in Comm.*, vol. 3, no. 3, pp. 457-567, May 1985.

[4] J.W. Wong, "Broadcast Delivery," *Proc. IEEE*, vol. 76, no. 12, pp. 1,566-1,577, Dec. 1988.

[5] T. Bowen, G. Gopal, G. Herman, and W. Mansfield, "A Scale Database Architecture for Network Services," *IEEE Comm.*, vol. 29, no. 1, pp. 52-59, Jan. 1991.

[6] S. Banerjee, V.O.K. Li, and C. Wang, "Distributed Database Systems in High-speed Wide-Area Networks," *J. Selected Areas in Comm.*, vol. 11, pp. 617-630, May 1993.

[7] T.S. Yum and K.H. Yeung, "The Confirm Before Delivery Technique for High Speed Data Distribution," *Proc. GLOBECOM'93*, pp. 1,105-1,109, Nov. 1993.

[8] W. Feller, *An Introduction to Probability Theory and Its Applications*, third ed., vol. I, p. 60. New York: John Wiley and Sons, 1968.

[9] R.J. McEliece, *The Theory of Information and Coding*. Reading, Mass.: Addison-Wesley, 1977.

[10] G.K. Zipf, *Human Behaviour and the Principle of Least Effort*. Reading, Mass.: Addison-Wesley, 1949.

[11] C. Ruemmler and J. Wilkes, "An Introduction to Disk Drive Modeling," *Computer*, vol. 27, no. 3, pp.17-28, Mar. 1994.