A Simple Analysis of Average Queueing Delay in Tree Networks

Eytan Modiano, *Member IEEE*, Jeffrey E. Wieselthier, *Senior Member IEEE*, and Anthony Ephremides, *Fellow IEEE*

***Abstract*—We develop an approach to the analysis of average queueing delay in a tree network of discrete-time queues with constant service time. The analysis of such systems is pertinent to packet-switched data networks with fixed-length packets. Our solution is based on considering an equivalent network, in which at each node packets in transit are given priority over exogenous arrivals. The solution to the equivalent model is easily computed, and, hence, the solution to the original model can be obtained.**

***Index Terms*— Queueing networks, tandem queue, queueing delay, equivalent model, priority disciplines.**

I. Introduction

In this correspondence we consider a network of discrete-time queues in which the service time is deterministic and is the same at each queue. Such a network of queues arises in a data communication network model, where data is formatted into fixed-length packets. An important performance index in such networks is average queueing delay. The model generally used for the analysis of average delay in a large network is *Kleinrock's Independence Assumption*, under which the queues at each link behave as independent queues regardless of the interaction of traffic between the different links [1], [2]. This model is reasonably good for systems involving exponential arrivals, a densely connected network, and uniform loading among source–destination pairs. Otherwise, the model may be very inaccurate in predicting delay. There have also been numerous other approximations proposed that have had varying degrees of success.

The exact computation of average delay in a network of queues with constant service time is complicated by the fact that the outputs of intermediate queues in the network (which serve as inputs to other queues) are correlated. This correlation, in an N-node network, renders the N-dimensional embedded infinite Markov chain that models the system intractable. There has been some effort in the literature to develop a delay model for a packet-switched network with fixed-length packets. It is clear that in a discrete-time network with fixed-length packets of one-slot duration, only one packet can depart from a link during any slot. In [3] an attempt is made to take advantage of this regular nature of departures by modeling the departures from a link in the network as an independent Bernoulli process with rate equal to the utilization of that link. In [4] an approximate model for average delay is developed which assumes the departures from intermediate links to be second-order Markov; with proper choice of parameters, it is then possible to obtain a fairly good approximation for the average delay in a linked-cluster architecture network. Perhaps the most promising approach is that of [5], where two discrete-time queues are considered in tandem with identical service time; by the use of direct calculations the moment generating function for the steady-state queue sizes can be obtained. This is the only exact solution, that we know of, to the problem of two queues in tandem with exogenous arrivals at both queues. However, the direct nature of this approach makes it difficult to extend to more than a simple tandem because of the rapidly increasing complexity of the calculations. There are also models for delay analysis of simple queue tandems [6], [7], where exogenous arrivals are not allowed at the second queue. In [8], a simple formula is derived for mean delays in multiplexers, which perhaps may be applicable to extensions of this problem, and in [9] a solution is developed for a fluid analog of the tandem queue model where the inputs are independent Levy processes.

Manuscript received August 31, 1994; revised July 31, 1995. The material in this correspondence was presented in part at the IEEE International Symposium on Information Theory, Trondheim, Norway, June 1994. This work was performed while E. Modiano was an NRC Post-Doctoral Research Associate at the Information Technology Division of the Naval Research Laboratory.

E. Modiano is with the Communications Division of MIT Lincoln Laboratory, Lexington, MA 02173 USA.

J. E. Wieselthier is with the Information Technology Division, Naval Research Laboratory, Washington, DC 20375 USA.

A. Ephremides is with the Department of Electrical Engineering and the Institute of Systems Research, University of Maryland, College Park, MD 20742 USA.

Publisher Item Identifier S 0018-9448(96)01029-2.

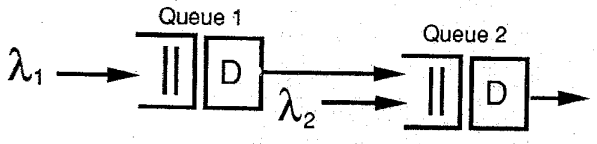

Fig. 1. Two discrete-time queues in tandem.

In this correspondence we offer an exact solution to tree networks of discrete-time queues with identical service times. Although the topology of such networks is restricted, our method illustrates how the bottleneck of interdependencies among interconnected queues can be broken and may inspire a solution to the case of completely general topology. Our solution is based on considering an equivalent network, where priority is given to customers in transit (over exogenous customers entering the system). The solution to the equivalent model is easily computed; from this the solution to the original model (without priorities) can be obtained. The priority scheme is only devised as a means to simplify the analysis of the system, and is not proposed for actual use in a network. For the purpose of demonstrating our approach, we begin by deriving the results for a tandem, and then we show how this approach can be extended to a tree network. Our hope is that this new approach using hypothetical priority disciplines will lead to the solution of more general systems of queues with wider applicability.

II. Two Queues in Tandem

Here we consider two discrete-time queues in tandem as shown in Fig. 1. The input to the first queue is a Poisson process of rate λ_1, and its output is fed into the second queue. Additionally, the second queue has an independent exogenous Poisson arrival stream of rate λ_2. Both queues have the same, constant, service time. Such a situation may arise in a packet-switched network where the packets are of a fixed size and take a constant amount of time to be transmitted. For simplicity we assume the service time to be one time unit, and that arrivals take place at the end of these unit time intervals. This assumption is necessary in order to maintain synchronization within the system. Without this assumption packets arriving to the system earlier in the unit interval would have to wait to the end of the interval before entering the system. On average this would result in additional delay of one half of a time unit [11]. Thus the arrival process can be

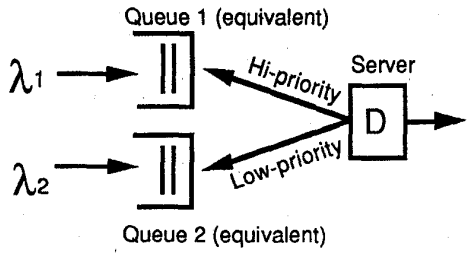

Fig. 2. An equivalent system with priority.

viewed as offering i.i.d. Poisson arrivals at each integer time instant. This simplification is nonessential but facilitates the analysis.

The first queue can then be viewed as a simple discrete-time M/D/1 queue with arrival rate λ_1. The average queue size for such a system is well known and is equal to $\lambda_1^2/2(1-\lambda_1)$ [11]. This result, which applies to continuous systems, is valid for discrete-time systems as well if all arrivals occur at the end of the time intervals, as we have assumed. By Little's formula the average queueing delay can then be expressed as $\lambda_1^2/2\lambda_1(1-\lambda_1)$. Since the service time at either queue is constant and equal to one time unit, the total wait (service time plus queueing delay) at the first queue is equal to $\lambda_1/2(1-\lambda_1)+1$.

The analysis of average delay at the second queue is complicated by the fact that the inputs to that queue are a mixture of the outputs of the first queue and an exogenous Poisson stream. Clearly, the second queue is not M/D/1. The direct solution to the average delay at the second queue is complicated because it amounts to analyzing a two-dimensional infinite Markov chain of general structure. In [5], a direct approach is used to solve for the average delay at the second queue using a moment-generating function approach. Here we offer a much simpler solution, which we easily extend to more complicated systems.

Let us assume that in-route traffic has priority over exogenous traffic. That is, customers departing from the first queue are immediately served at the second queue without experiencing any delay. This assumption is not required of the actual system, and is only a hypothetical one made for the purpose of simplifying the analysis. In fact, we note that the overall average delay at the second queue is the same for either the system with priority or the actual system without priority. It is known that assigning priority to any type of traffic does not alter the average delay in the overall system, so long as the priority is not assigned on the basis of the required service time [2]. In this system, where all customers require the same service time, the priority assumption does not alter the overall queue size; therefore, the average delay for the system without priority is the same as for the system with priority.

Now considering the tandem with priority, we observe that customers departing from the first queue will experience no delay at the second queue. This is easily seen as a consequence of the discrete-time nature of the system. Departures from the first queue (which all have high priority) will be available for transmission only at time-unit boundaries, and hence will not have to wait for the completion of a customer whose service is currently in progress in the second queue. Exogenous traffic arriving at the second queue will only be served when there are no customers departing from the first queue. Since a customer departs the first queue exactly one time unit after it begins receiving service, the entire two-stage tandem system can be thought of as a single-server system with two exogenous Poisson streams where one stream receives priority over the other. The equivalent system is shown in Fig. 2. Equivalence is in the sense of total average queueing delay (i.e., waiting time). Clearly, in Fig. 1 the traffic of the λ_1 stream incurs service time of one extra unit since it traverses two stages, while that of the λ_2 stream encounters a single-service stage. However, the queueing time (and queue size) is the same as the system of Fig. 2 because of the service synchronism. Of course, the delay in this equivalent system accounts for the queueing delay at both queues. For the tandem with priority, in-route customers will experience all of the queueing delay at the first queue and no queueing delay at the second queue. The delay experienced by exogenous traffic at the second queue is the same as the delay experienced by the low priority traffic in the equivalent system of Fig. 2. It is solved as follows:

First, we define the following quantities relating to the equivalent system.

- $\overline{Q}\Delta$ the average overall queue size (the sum of the low and high priority queues).
- $\overline{Q}_1\Delta$ the average queue size at the high-priority queue.
- $\overline{Q}_2\Delta$ the average queue size at the low-priority queue. (This quantity is equal to the average queue size for the exogenous traffic in the tandem network with priority).

Thus $\overline{Q}_2 = \overline{Q} - \overline{Q}_1$. Now, $\overline{Q}$ for the equivalent system with priority is the same as that of a system without priority. It is simply the average queue size for an M/D/1 queue with arrival rate $\lambda_1+\lambda_2$, which is equal to $(\lambda_1+\lambda_2)^2/2(1-\lambda_1-\lambda_2)$. Also, $\overline{Q}_1$ in the system with priority is the same as the average queue size for the first queue in the original system and is equal to $\lambda_1^2/2(1-\lambda_1)$. So now we can express the average queue size in the second queue as follows:

$$\overline{Q}_2 = \frac{(\lambda_1+\lambda_2)^2}{2(1-\lambda_1-\lambda_2)} - \frac{\lambda_1^2}{2(1-\lambda_1)}.$$

Returning to our tandem, in which priority is given to in-route traffic, the second queue will consist only of exogenous customers. However, the average queue size at the second queue is the same for both the system with and without priority, and therefore the average queueing delay $\overline{W}_2$ (for both in-route and exogenous traffic) at the second queue of the tandem can now be expressed using Little's formula as

$$\overline{W}_2 = \frac{\overline{Q}_2}{\lambda_1+\lambda_2}.$$

Thus the total wait $\overline{D}_2$ (which includes the service time) is equal to

$$\overline{D}_2 = \frac{\overline{Q}_2}{\lambda_1+\lambda_2} + 1.$$

This result is consistent with that obtained in [5] when the arrival processes are independent Poisson processes.

It is also interesting to compare this result to that obtained using an independence approximation in which the average queue size at the second queue would be given by the formula for an M/D/1 queue with arrival rate $\lambda_1+\lambda_2$ and would equal $(\lambda_1+\lambda_2)^2/2(1-\lambda_1-\lambda_2)$. Comparing this to the exact result obtained above it is clear that when the "in-route" traffic rate is substantially greater than that of the exogenous traffic, the independence approximation is inaccurate in predicting delays.

III. Extension to Larger Systems

In the previous section we looked at an equivalent system with priority, which simplified our analysis by eliminating the delay experienced by "in-route" traffic in the second stage of the tandem. That delay is the major source of difficulty in analyses of this type because it depends on the joint behavior at both service stages. In this section we use a similar technique to arrive at an average delay formula for a tree of queues of general structure such as the one shown in Fig. 3. Once again, we arrive at our solution gradually; as a first step we consider the two-level tree of queues shown in Fig. 4. In this system only the external queues have exogenous arrivals. The

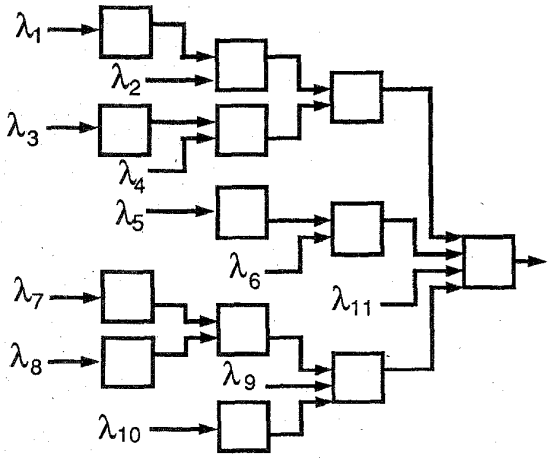

Fig. 3. A concentrating tree of discrete time queues.

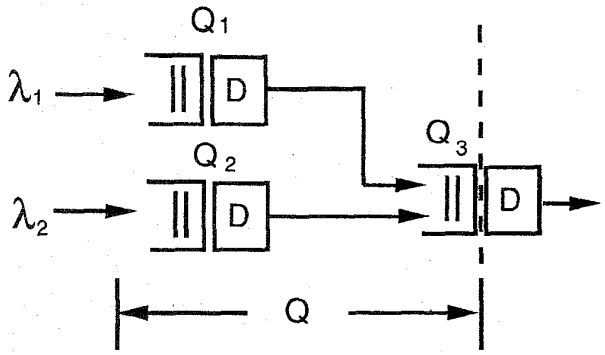

Fig. 4. A two-level tree.

"root" queue has the superposition of the outputs of all the other queues as its input.

The analysis of average delay in this system is fairly simple. We begin by defining the following quantities:

- $Q_i(t)\triangleq$ the total number of customers present in Q_i during slot t (here we include all customers, whether waiting in queue or being served), $i = 1, 2$.
- $Q_3(t)\triangleq$ the number of customers waiting in Q_3 during slot t (here we do *not* include customers being served).

Now let

$$Q(t) = Q_1(t) + Q_2(t) + Q_3(t)$$

which denotes the number of customers in the entire system, except those in service at the head queue.

Next we write the transition equations for each of the queues, that is

$$Q_1(t) = Q_1(t-1) + A_1(t-1) - I(Q_1(t-1)) \quad (1)$$

$$Q_2(t) = Q_2(t-1) + A_2(t-1) - I(Q_2(t-1)) \quad (2)$$

and

$$Q_3(t) = Q_3(t-1) + I(Q_1(t-1)) + I(Q_2(t-1)) - I\Big(Q_1(t-1) + Q_2(t-1) + Q_3(t-1)\Big) \quad (3)$$

where $A_i(t)$ is equal to the number of new arrivals to queue i during slot t and $I(x)$ is equal to 1 if x is greater than 0, and 0 otherwise.

The first two equations are the standard transition equations for an M/D/1 queue. The last equation requires explanation. Here we recall that $Q_3(t)$ does not include those customers currently being served by queue 3. Therefore, if a customer arrives (from either queue 1 or queue 2) to an empty queue 3, it immediately enters the server and Q_3 remains empty. However, if Q_3 is not empty, the first customer in Q_3 enters the server; Q_3 is decremented by 1 and incremented by the number of new arrivals from queue 1 and queue 2.

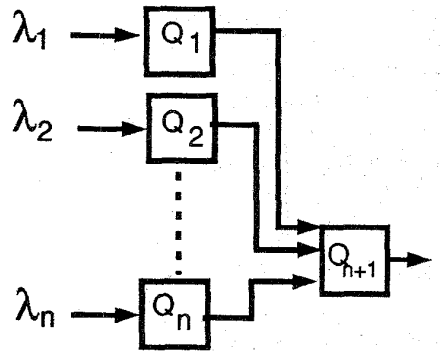

Fig. 5. An extension to the two-level tree.

Now, by adding (1)–(3) and by definition of $Q(t)$ we obtain,

$$Q(t) = Q(t-1) + A(t-1) - I(Q(t-1)) \quad (4)$$

where $A(t) = A_1(t) + A_2(t)$. Now we notice that (4) is identical to (1) with Q replacing Q_1 and A replacing A_1; that is, (4) is the transition equation for an M/D/1 system. Hence, since $\overline{A}(t) = \lambda_1 + \lambda_2$, we obtain

$$\overline{Q} = \frac{(\lambda_1+\lambda_2)^2}{2(1-\lambda_1-\lambda_2)} + \lambda_1 + \lambda_2.$$

Since $Q_3 = Q - Q_1 - Q_2$ and since Q_1 and Q_2 are both queue sizes of M/D/1 systems with arrival rates λ_1 and λ_2 respectively, we have

$$\overline{Q}_3 = \frac{(\lambda_1+\lambda_2)^2}{2(1-\lambda_1-\lambda_2)} - \frac{\lambda_1^2}{2(1-\lambda_1)} - \frac{\lambda_2^2}{2(1-\lambda_2)}.$$

It is now straightforward to extend the approach to an arbitrary number of M/D/1 queues feeding into one queue with an infinite buffer. Consider the system of Fig. 5, where n M/D/1 queues with arrival rates $\lambda_1 \cdots \lambda_n$ feed into one queue. Then, $\overline{Q}_{n+1}$, the average queue size at that queue can be written by inspection as follows:

$$Q_{n+1} = \frac{\left(\sum_{i=1}^{n}\lambda_i\right)^2}{2\left(1-\sum_{i=1}^{n}\lambda_i\right)} - \sum_{i=1}^{n}\frac{\lambda_i^2}{2(1-\lambda_i)}. \quad (5)$$

In order to proceed to our next building block in our development of a solution to the tree network, we must use a lemma from Morrison [10]. This lemma relates the output of the "root" queue in a tree of queues to that of a single M/D/1 queue with appropriate inputs. The lemma essentially states that given a concentrating tree of discrete time queues with identical service times and independent arrivals to each of the queues, the output of the "root" queue is the same as the output of a single queue with the same service time and with an arrival pattern that results from the superposition of the individual arrival processes that feed into the "root" queue.

Because in our case we are interested in Poisson arrival streams, this lemma implies that the single queue with identical output stream as the root queue will be M/D/1. Thus the output of the root of the tree in Fig. 3 is statistically the same as the output of the M/D/1 queue shown in Fig. 6, with arrival rate equal to the sum of the arrivals into the tree of Fig. 3. This does *not* imply that the queue at the equivalent

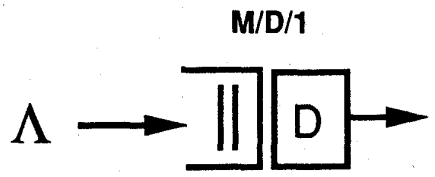

Fig. 6. An equivalent M/D/1 queue to the tree of Fig. 3 $\left(\Lambda = \sum_{i=1}^{i=11} \lambda_i\right)$.

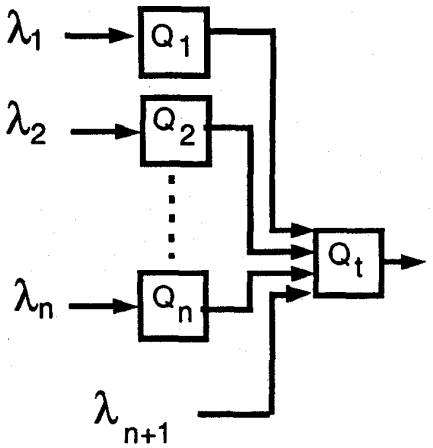

Fig. 7. A two-level tree with an exogenous input.

M/D/1 queue has the same behavior as the queue at the root of the tree of queues in all respects. It simply agrees with respect to the output process. Also, a direct consequence of this lemma is that the state of the server (busy or idle) will have the same statistics as the server of the corresponding M/D/1 queue.

Now consider the two-level tree in Fig. 7. This tree is the same as the one in Fig. 5, with the addition of an exogenous Poisson source of rate λ_{n+1}. The solution for the queue size at the root of this tree is now trivial with the use of the lemma and the priority technique of Section I. Suppose, as in Section I, that we give priority to "in-route" traffic over the exogenous traffic. Then we know using the lemma from [10], that queue Q_t is occupied with "in-route" (high priority) traffic as a single M/D/1 queue with arrival rate

$$\lambda = \sum_{i=1}^{i=n} \lambda_i.$$

Now, the exogenous traffic will only be served when the server at Q_t is idle. To characterize the exogenous traffic, we consider a priority M/D/1 queue which consists of two input streams, i.e., a high-priority stream of rate λ and a low-priority (exogenous) stream of rate λ_{n+1}. We know from Section I that the average queue size, $\overline{Q}_e$, for the low-priority (exogenous) traffic in this case would be

$$\overline{Q}_e = \frac{(\lambda + \lambda_{n+1})^2}{2(1 - \lambda - \lambda_{n+1})} - \frac{\lambda^2}{2(1-\lambda)}.$$

Finally, the total queue size at the root node, $\overline{Q}_t$, is the sum of the exogenous queue size and the queue of "in-route" customers. The latter quantity is equal to $\overline{Q}_{n+1}$ from (5). Summing the two equations we have,

$$\overline{Q}_t = \frac{(\lambda + \lambda_{n+1})^2}{2(1 - \lambda - \lambda_{n+1})} - \sum_{i=1}^{i=n} \frac{\lambda_i^2}{2(1-\lambda_i)}. \tag{6}$$

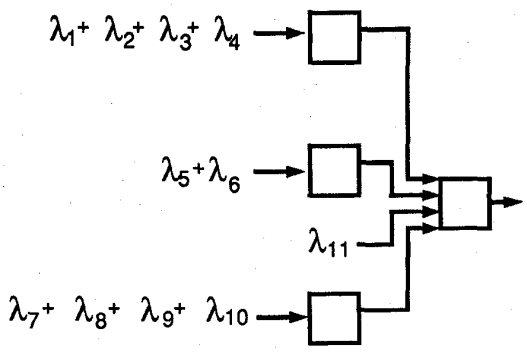

Fig. 8. An equivalent two-level tree to the tree of Fig. 3.

By invoking now the lemma of Morrison [10] discussed earlier, we can solve for the average queue size at the root of any concentrating tree of queues. As observed above, any rooted tree of queues can be replaced by an equivalent two-level tree with one exogenous source and an arbitrary number of internal queues representing the different branches of the tree. Each of these internal queues will have the same output as the tree branch that it represents; therefore, the queue size at the root queue would be the same for both the original tree and its equivalent two-level tree. The solution for the queue size of the two-level tree is obtained using (6). For example, consider the rooted tree of Fig. 3. Its equivalent two-level tree is shown in Fig. 8. The average queue at the root node is given by

$$Q_R = \frac{\left(\sum_{i=1}^{i=11} \lambda_i\right)^2}{2\left(1 - \sum_{i=1}^{i=11} \lambda_i\right)} - \frac{\left(\sum_{i=1}^{i=4} \lambda_i\right)^2}{2\left(1 - \sum_{i=1}^{i=4} \lambda_i\right)} - \frac{\left(\sum_{i=5}^{i=6} \lambda_i\right)^2}{2\left(1 - \sum_{i=5}^{i=6} \lambda_i\right)} - \frac{\left(\sum_{i=7}^{i=10} \lambda_i\right)^2}{2\left(1 - \sum_{i=7}^{i=10} \lambda_i\right)} \tag{7}$$

and the average queueing delay, using Little's formula, is given by

$$\overline{W}_R = \frac{\overline{Q}_R}{\sum_{i=1}^{11} \lambda_i}.$$

Finally, the total time, including the time spent in the server, is

$$\overline{D}_R = \overline{W}_R + 1.$$

Notice, again, that if an independence approximation were used to obtain the queue size at the root, the resulting queue size would be that of an M/D/1 queue with the same total arrival rate, and would be equal to the first term in (7). Therefore, at least when the system is lightly loaded, it is apparent that the independence approximation is inaccurate in predicting delay. However, when the system is heavily loaded, the first term of (7) dominates the other terms and the independence approximation may yield results with improved accuracy.

IV. Conclusion

We have obtained an exact solution for the average delay in a tree network of discrete-time queues where the service time is constant and identical at every queue. While in general the solution for the

average delay in a network of queues is difficult to obtain, here we were able to use the fact that the service time is a constant to obtain the desired result. Our approach utilized i) a simple hypothetical priority discipline as a "shortcut" and ii) the queue-equivalence lemma of Morrison [10]. This result is applicable, in particular, to a packet network where data packets are of fixed duration. Of course, most networks do not have a simple tree topology and involve in general a more complicated structure, where some packets are allowed to by-pass certain nodes and some packets exit the system before reaching the root of the tree. The solution developed here does not apply to these cases.

A useful extension of our approach would be along the lines considered in [3], which addresses a network of discrete-time queues where the different queues have constant, but different service times. Such a system, again, arises in a packet radio network where the data capacity of the network varies from link to link. The solution for the delay in such systems would be very useful because it would permit the exact computation of delay for more realistic network models and would facilitate the solution to the optimal capacity allocation problem. In fact, an approximate solution for the case of unequal capacities was obtained in [3] by modeling the output of each link as Bernoulli with rate equal to the utilization of that link. The results of this approximation compared favorably with simulation, especially for large networks.

REFERENCES

[1] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*. New York: McGraw-Hill, 1964.

[2] D. P. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987.

[3] E. Modiano, J. Wieselthier, and A. Ephremides, "An approach for the analysis of packet delay in an integrated mobile radio network," in *Proc. 1993 Conf. on Information Science and Systems*, (Baltimore, MD, Apr. 1993), p. 138.

[4] A. M. Viterbi, "Approximate analysis of time-synchronous packet networks," *IEEE J. Sel. Areas Commun.* vol. SAC-4, pp. 879–890, Sept. 1986.

[5] J. A. Morrison, "Two discrete-time queues in tandem," *IEEE Trans. Commun.* vol. COM-27, pp. 563–573, Mar. 1979.

[6] O. J. Boxma, "On a tandem queueing model with identical service times at both counters, I," *Adv. Appl. Prob.*, vol. 11, pp. 616–643, 1979.

[7] ——, "On a tandem queueing model with identical service times at both counters, II," *Adv. Appl. Prob.*, vol. 11, pp. 644–659, 1979.

[8] H. Dupuis and B. Hajek, "A simple formula for mean delay for independent regenerative sources," *Queueing Syst.*, vol. 16, pp.195–239, 1994.

[9] O. Kella, "Parallel and tandem fluid networks with independent Levy inputs," *Ann. Appl. Prob.*, vol. 3, pp. 682–695, 1993.

[10] J. A. Morrison, "A combinatorial lemma and its application to concentrating trees of discrete-time queues," *Bell Syst. Tech. J.*, vol. 57, no. 5, pp. 1645–1652, May–June 1978.

[11] G. D. Stamoulis and J. N. Tsitsiklis, "Efficient routing schemes for multiple broadcasts in hypercubes," Rep. LIDS-P-1948, Laboratory for Information and Decision Systems, MIT, 1990.

Hashing of Databases Based on Indirect Observations of Hamming Distances

Vladimir B. Balakirsky

***Abstract*—We describe hashing of databases as a problem of information and coding theory. It is shown that the triangle inequality for the Hamming distances between binary vectors may essentially decrease the computational efforts of a search for a pattern in a database. Introduction of the Lee distance in the space, which consists of the Hamming distances, leads to a new metric space where the triangle inequality can be effectively used.**

***Index Terms*—Hashing, searching for patterns, coding, decoding.**

I. INTRODUCTION

One of important problems in computer science can be represented as follows: we have given a collection of items, and we wish to store these items and upon demand retrieve the items whose key values match given key values. A particular approach to the storage and retrieval problem is known as hashing when we use the key value of an item to compute an address for the storage of the item. Since the mapping between keys and addresses is not one-to-one, the events when different keys have the same address may take place; these events are known as collisions, and their resolution is of the main interest for a hashing scheme [1]–[4].

Similar ideas form the basis for the procedures which are referred to as external hashing [5]. It is known that the cost per storage increases as the access time decreases. Main memories usually have random access time. Since their size is limited by the cost requirements, databases are stored in the secondary memory with a rather slow access [6]. The hashing technique can essentially reduce the total number of accesses required if the values of a hash function applied to the keys of each item of the database are stored in the main memory. To find the item with a certain key we calculate the value of the hash function for that key and access only the items whose key values have the same value of the hash function. In the present correspondence, we address the external hashing and formulate the following problem (to simplify formalization, the item and its key value are associated): we are given a collection of items which are binary vectors of the same length stored on the external memory, and a part of random-access memory (RAM) that can be filled with the values of a hash function corresponding to each item. For a given pattern, which is a binary vector of the same length as the items, we must find all the items located at the Hamming distance of less than a fixed threshold value.

We develop the external hashing in two directions. First, the value of the threshold can be positive, which puts some restrictions on the hash functions allowed. This is because we want to get information about the Hamming distances between binary vectors based on the comparison of the values of the hash function corresponding to these vectors. To make it possible, we use the metric properties of the Hamming space and extend some approaches by Koshelev [7] to the

Manuscript received February 10, 1994; revised October 20, 1995. The work was supported by a Scholarship from the Swedish Institute, Stockholm, Sweden.

The author is with the Department of Information Theory, University of Lund, P.O. Box 118, S-221 00 Lund, Sweden ,on leave from The Data Security Association "Confident," 193060 St. Petersburg, Russia.

Publisher Item Identifier S 0018-9448(96)01479-4.