J.King Saud Univ., Vol. 9, *Computer & Information Sci.*, pp. 95-123 (A.H 1417/1997)

Improving Fairness in Packetized Computer Data Networks

Gamal Abdel-Fadeel Mohammed
Faculty of Engineering & Technology, Communications Department, Helwan University, Helwan, Egypt

(Received, 12 February 1995; accepted for publication, 4 June 1995)

Abstract. The use of transmission scheduling to improve a measure of fairness in packet data networks is investigated. This fairness measure is based on mean end-to-end delays derived from Kleinrock's classical model. It is found that reallocating delay among user classes in terms of their delays incurred while routing through the network can be used to improve the fairness of a network. In that respect, a model for estimating the packet's delay is formulated. This model is used as the basis in developing a dynamic priority scheme which is aimed at redistributing the queuing delays of the packets against variations in the offered traffic to the network. The operation and performance of the above system is tested in terms of simulation study, and results obtained show the effectiveness of the new approach in equalizing the mean end-to-end delays of different packet classes in the network. Thus improving the fairness of the network.

1. Introduction

In recent years, we have seen increased research activities directed towards the design and performance evaluation of packet-switched networks. The delay and throughput characteristics under various operational strategies have been investigated. These strategies include routing [1-2], flow and congestion control [3-4], and buffer management [5-6]. Studies on network design have also resulted in the formulation and solution of a number of optimization problems, mainly with the minimization of delay or cost [7-8]. Two issues, however, affecting the design and performance of packet-switched networks seem to have received less attention. These are the effects of scheduling and the evaluation of fairness as a performance measure.

Studies on fairness are promoted by the fact that network design algorithms are not inherently fair. These techniques usually attempt to optimize some average parameters, like total throughput or network delay. It is well known, however, that the optimum solution produced may favor some users and give others a significantly bad performance (examples have been given [9], where the optimal solution yields throughput zero for some users).

Some research attention [9-12], has been devoted to the issue of fairness. They may be categorized into two lines of approach.

One relates the throughput characteristics under various routing and flow control strategies to fairness. In this approach, fairness is defined as "uniform throughputs" among network users. It includes Gerla and Staskausks [10]. Thaker and Cain [11] and Bharath-Kumar and Jaffe [9]. The results from the former two studies [10-11] is that fairness can be achieved by simply adjusting routing and flow control parameters. Bharath-kumar and Jaff's study [9], on fairness is mainly focused on the relationships between several versions of power (throughput divided by mean delay) and fairness. One of their findings on fairness is that some versions of power are unfair in a loose sense so that power-oriented flow control schemes may result in an unfair throughput allocation.

The other approach by Wong et al. [12], relates the relative delays among users, or equivalently packet classes, under various priority scheduling strategies to fairness. In their analysis, a fairness measure, the square coefficient of variation of the individual class delays, was devised and it was shown that scheduling strategies can be used to improve the fairness considerably. Their analysis was based on the observation that under the Poisson assumption on the arrival process of packets the selection of scheduling disciplines would not significantly affect the mean end-to-end (ETE) delay over all classes of packets.

This paper addresses the fairness issue by relating the mean ETE network delays to the subscriber attributes namely, their physical location and offered load. The approach taken is to redistribute the overall network delay among packets belonging to different subscribers so that all subscribers experience the same mean ETE delay irrespective of their physical locations on the network. This is achieved by varying the transmission scheduling of the packets at the output channel. We propose an on-line delay estimation algorithm to be used in conjunction with the transmission scheduling scheme. This has resulted in a dynamic priority system which is capable of reliably tracking the current state of the network delays as it changes with time.

The delay estimation algorithm developed here is based on the theory of statistical inference for point processes. This body of material is known as the martingale method; see for example, [13-14] and [15]. This method of statistical inference pertains to situations where observations are obtained from a point process (or perhaps several point processes) together with partial information about an associated stochastic process which has unknown parameters, or functions. The basic idea behind the method is that, for a process which cannot be computed from the observations, an estimate of the process can be made using a process that differs from the process under examination by a martingales. This concept is known as the martingales representation theorem [16].

We consider an environment where the network topology and the traffic matrix are given. The packet-switched network model is based on the Kleinrock's classical model for packet networks [17], with the mean ETE delay as its basic performance measure.

2. Network Model and Assumptions Concerning Terminal Traffic

The packet data network considered here is composed of an N switching nodes linked by L unidirectional communication links. As shown in Fig.1., each node services a number of terminals connected to the node via local loops of finite speed and a device which collects a certain number of bits of incoming messages into packets of fixed length. Due to the finite speed of the local lines, messages arrive at the packet assembly circuit not instantaneously but more or less dispersed in time. We assume that all terminals generate messages at the same average rate and that the time between consecutive messages are exponentially distributed. The message lengths are assumed to be independent and discrete so that all messages can be broken down into an integer number of packets at the assembly devices (the problem area of incompletely filled packets will not be considered here).

We take sufficiently many terminals to justify the assumption that the overall message generation is a Poisson process. Following Rudin's reasoning [18], and the derivation provided by Lewis [19], for a relatively low speed local loop, the distribution of the interarrival time between packets indicates that packets arrive at the server (see

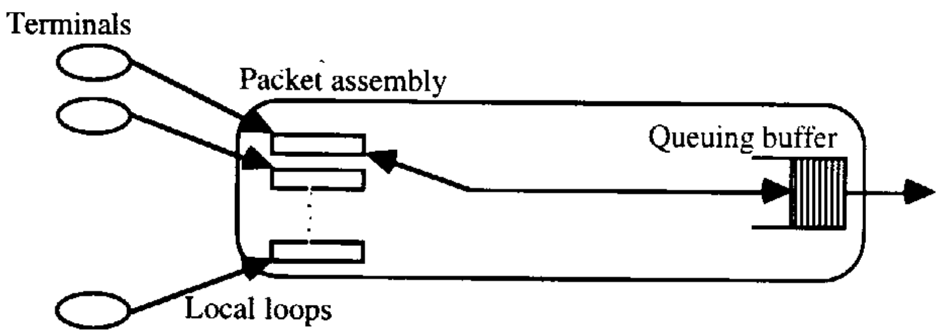

Fig. 1. A packet switching node with output queuing buffer.

Fig. 1.) in a Poisson fashion with arrival rate depending on the speed of the local loops and on the call (message) rate which in turn depends on the number of active terminals. As in Kleinrock's classical model, the delay experienced by a given packet can be approximated by the queuing time and the data transfer time in the links. The processing time at the switching nodes and the propagation delays are assumed to be negligible. In the network, packets are classified according to source-destination pairs. In particular, a packet is said to belong to class (s, d), if its source is s and its destination is d. For convenience, it is assumed that the packet classes are numbered from 1 to R, and r is used instead of (s, d) to denote a packet class.

3. Upstream Packet-Delay Estimation Model

3.1 Characterizing the packetized data traffic

Consider the switching nodes in the network model described in Section 2 . Packets are sent to the (node) server when a virtual circuit has been set up. A virtual circuit is held for an exponentially distributed time with mean ω^{-1}, and the time between virtual circuit set-ups is also exponentially distributed with mean ζ^{1} . It is seen that the packet arrivals to the server is a Poisson process, with rate depending on the rate of message generation at each terminal as well as on the number of active terminals. This means that the packet arrival process can be seen as a single Poisson arrivals (from a single terminal) modulated by the number of active terminals which in itself is a stochastic process. Therefore, a simple Poisson arrival may not be a good approximation for the overall packet arrivals at a given node. Arrival streams of this type were treated in Heffes [20], and Burman and Smith [21]. According to Heffes, the switching node is considered to be connected to an N fixed terminals. Each of these terminals is in one of two states, active or inactive. Assume that a terminal is in these states for an exponentially distributed amount of time with mean α^{-1} and β^{-1}, respectively. Hence, the overall packet arrivals is seen as a Poisson process with rate equal to $\lambda_{terminal}$ time the number of the active terminals. The variations in the number of active terminals is modeled as a Markov process with each of the (Markov) state determining the actual number of active terminals at a given time t. Therefore, the overall arrival process is considered as a Markov Modulated Poisson Process (MMPP).

Consider a packet-switched network with each of its (node) servers constructed as described in Section 2. In the network, packets belonging to different classes mix together in the queuing buffers as they are routed toward their destinations. Our objective, now, is to quantify the effect of the traffic mixing (interference) process on the mean interarrival time of a given packet class along their route to the destination node.

Assume a queuing buffer where a set of traffic classes share a common output buffer, as illustrated in Fig. 2. For a given class i packets, let :

C_n^i : denotes the n^{th} packet.

τ_i : denotes the arrival time.

t_n^i : denotes the n^{th} departure time.

W_n^i : denotes the waiting time for the n^{th} packet.

X_n^i : represents the time taken to serve (transmit) the n^{th} packet.

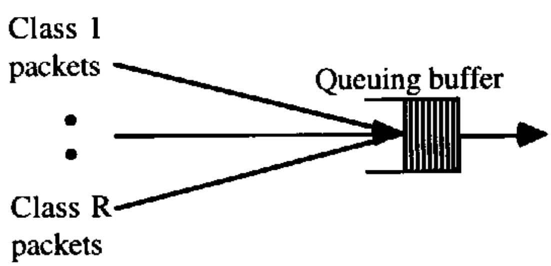

Fig. 2. Different traffic classes sharing a common output buffer.

S_n^i : denotes the response time for the n^{th} packet.

T_n^i : denotes the interarrival time for the n^{th} packet.

$\overline{T}_n^i$: denotes the interdeparture time for the n^{th} packet.

From the timing diagram shown in Fig. 6, the average number of packet arrivals during the period T_n^i can be written as,

$$K = \lambda \quad T_n^i \ , \tag{1}$$

where

$$\lambda = \sum_{\substack{r \varepsilon R \\ r \pi i}} \lambda_r \tag{2}$$

Here, R denotes the set of all classes sharing the common (output) buffer, and λ_r is the arrival rate of class r measured in packets/sec. From Fig. 6, the interdeparture time, $\overline{T}_n^r$ can be determined as,

$$\overline{T}_n^r = \sum_{j=1}^{K} X_n^j + X_{n+1}^r, \tag{3}$$

where K is given by equation (1). Assume that the average service time for an arbitrary class r packet is $\overline{X}$. Then $\overline{T}_n^r$ can be determined as follows,

$$\overline{T}_n^r = (K + 1)\,\overline{X}. \tag{4}$$

on using (1), it can be seen that,

$$\overline{T}_n^r = (\lambda\, T_n^r + 1)\,\overline{X}. \tag{5}$$

Let ρ_r be the traffic intensity of class r then,

$$\begin{aligned} \rho_r &= \lambda_r \overline{X} \text{ and.} \\ \rho &= \lambda \overline{X}, \end{aligned} \tag{6}$$

where ρ is the intensity of all the traffic classes interfering with class r at a given node. Therefore, when using equation (6) in (5), the result

$$\overline{T}_n^r = \rho \ T_n^r + \overline{X} . \tag{7}$$

is obtained. Equation (7), represents a quantification for the effect of traffic interference on the interdeparture time of a given class r packets while traversing the network.

From the findings presented above, it is easy to see that the interdeparture time of a given class r packets is, in effect, modulated by two factors:

a) The intensity of the interfering traffic while traversing the network.
b) The number of active terminals at the source node and the virtual circuit (call) attempt process.

Fortunately, both of the above two factors modulate the packet's interdeparture time in the same manner. In the rest of this paper we assume that the (modulation) effect of the number of active terminals in b) is dominated by the effect of the interfering traffic intensity along the path down to the destination nodes.

Consider a given packet network, where a given node j is downstream with respect to another node i. Clearly, the interdeparture time of the packets transmitted from node i to node j is seen as an interarrival time at node j. In the delay estimation model being developed the interarrival time for a given class r packets is considered as the, only, observations available for the delay estimation scheme. Moreover, these observations are looked on as a time-point process which has its interarrival time modulated by a finite-state Markov process (i.e., Markov Modulated Point Process, MMPP). The states of the (modulating) Markov process is the number of active terminals and is refered to as the interfering traffic intensity in the network.

3.2 Recursive Estimation From Discrete-Time Point Process

Assume that the time is divided into intervals of length Δt and consider the sequence of observations $\{n(t), t = 1, 2, \ldots\}$, being simply a binary sequence describing the occurrences of some (arrival) events. Here $\{n(t) = 1\}$ shows that a packet arrived at time t, and $\{n(t) = 0\}$ shows that there is no arrival at time t. Suppose the probability of occurrence at a given time is affected by both the previous occurrences as well as by some other related sequence $\{x(t)\}$. Here x(t), represents the state of the interfering traffic intensity. Assume that the Markov process has the following state space,

$$\gamma^0 < \gamma^1 < \gamma^2 < \ldots \gamma^N$$

where γ^k represents the rate (intensity) of the interfering traffic provided it is in state k. The state transition probabilities of the process x(t), can then be defined as follows,

$$Q(t) = \{q_{ij}\},$$

where

$$q_{ij}(t) = P_r \; (x(t+1) = \gamma^i \; / \; x(t) = \gamma^j)$$

Assume that the interfering traffic is generated from a single (big) group of terminals connected (through local loops) to one (big) node i. As shown in Fig. 3, the delay estimation scheme is implemented at node j which is downstream to node i. The observations seen by node j is the (time modulated) interarrival time of a given class r packets.

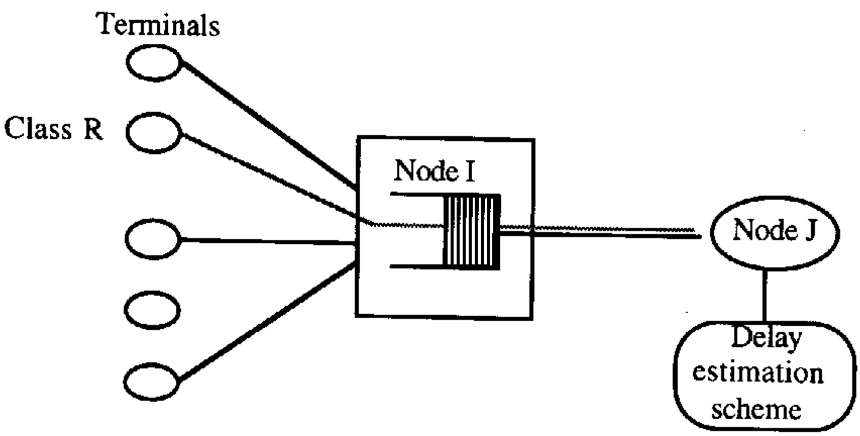

Fig. 3. Upstream-delay estimation scheme.

The, time-varying, number of active terminals at node i can be modeled [22], as a birth/death (B/D) process having N+1 states with B/D coefficients given by

$$\lambda(k,t) = \begin{cases} \lambda & \text{rate of transition from state k to k+ 1} \quad \forall k < N, \\ 0 & \forall k \geq N \end{cases}$$

$\mu(k,t) = k\mu$ rate of transition from state k to state k-1, k=0. 1,...N.

where as previously stated in Subsection 3.1,

$1/\lambda$ = mean inactive duration
$1/\mu$ = mean active duration

With this approximation, it is elementary that

$$\begin{aligned} q^T_{i,i}(t) &= (1-\lambda)(1-i\mu) + i\mu\lambda, \\ q^T_{i,1-i}(t) &= \lambda(1-i\mu), \\ q^T_{i,i+1}(t) &= i\mu\ (1-\lambda), \\ q^T_{i,j}(t) &= 0, \qquad \text{elsewhere,} \end{aligned}$$

Note that the argument t has been suppressed in both λ and μ .
Let us define

$$X_k(t) = \begin{Bmatrix} 1 \text{ if the interference process is in state } k, k=0, 1, \ldots, N \\ 0 \qquad \text{otherwise} \end{Bmatrix}$$

following Segall [23], we have (see Appendix A)

$$x(t+1) = Q^T(t)\, x(t) + u(t)$$
$$n(t) = \lambda^T(t)\, x(t) + w(t)$$

where $x^T(t) = \{x_0(t), x_1(t), \ldots, x_N(t), \}$.

$\lambda^T(t) = \{\lambda(0,t), \lambda(1,t), \ldots, \lambda(N,t)\}$ and u (t), w (t) are 'noise' porcesses. In effect, u(t) and w(t) are martingales difference sequences with respect to the σ-algebra generated by the observation sequences { n (0), n (1), . . . , n (t-1)} and {x (0), x (1), . . . , x (t)}. The following results are based on "Segall [23]: given the observation a time-point process, n (t), which is related to the interfering traffic x (t), the optimum one-step predictor is given by

$$\hat{X}(t+1/t) = Q^T(t)\hat{X}(t/t-1) + \frac{S^T(t)\hat{x}(t/t-1) - Q^T(t)\Sigma(t)\lambda(t)}{\lambda^T(t)\hat{x}(t/t-1) - (\lambda^T(t)\hat{x}(t/t-1))^2} .(n(t) - \lambda^T(t)\hat{x}(t/t-1)) \quad (8)$$

with $\hat{X}(1/0) = \pi(0)$ = a apriori probability distribution of the traffic interference process, $\Sigma(t) = \hat{x}(t/t-1)\,\hat{x}^T(t/t-1)$ and S is defined by

$$S_{ij}(t) = P_r[x_j(t+1) = 1, n(t) = 1/x_i(t) = 1],$$

with

$$S_{ij}(t) = i\mu\lambda$$
$$S_{i,i+1}(t) = \lambda(1 - i\mu)$$
$$S_{i,j}(t) = 0, \quad j \neq i, \; j \neq i+1$$

and

$$\hat{x}(t/t) = \hat{x}(t/t-1) + \frac{\text{diag}\{(\hat{x}(t/t-1))\} - \Sigma(t)\lambda(t)}{\lambda^T(t)\hat{x}(t/t-1) - (\lambda^T(t)\hat{x}(t/t-1))^2} (n(t) - \lambda^T(t)\hat{x}(t/t-1)) \quad (9)$$

where $\text{diag}\{\hat{X}(t/t-1)\} = \text{diag}\{\hat{X}_0(t/t-1), \ldots, \hat{X}_N(t/t-1)\}$.

Equations (8) and (9) represent an algorithm for a minimum error variance (delay) filter/predictor which are realized in the next Section.

4. Dynamic Priority Scheme for Data Packets

The objective of the dynamic priority scheme presented here is that users of public data networks should get uniform (fair) quality of service. Arguing in this direction, and considering user delays to be the most relevant performance measure. We choose to use the Kleinrock's time-dependent priority scheme [7] in scheduling the transmission of packets at the links. In the time-dependent priority scheme, the instantaneous priority of a packet belonging to a given class i traffic at time t is given by b_i (t - t_0) where t_0 is the packet arrival time to a given link and b_i is the (time-varying) priority level of class i packets. In this priority scheme, each node, l , is made to estimate the delay experienced by each class i packets along the path (route) from its source node down to itself (i.e., the upstream delay). This estimate is used to replace the instantaneous priority level, for class i packets. Assume that class i packets are to travel along the path shown in Fig. 4. Clearly, the (upstream) delay estimate made by node l is not the ETE delay used in most of the existing priority schemes. This implies that we are approximating the ETE delay by the (upstream) delay estimate. The justification of this approximation is given in the following.

Assume that the ETE delay along a given (route) path is constructed using the following equation

$$d_1^i = d_1^i + D_1^i + dd_1^i \tag{10}$$

where as shown in Fig. 4, d_1^i represents the estimate maintained by node l of the ETE delay of class i packets, du_1^i is the estimate maintained by node l of the overall delay of class i packets on the links upstream to node l, dd_1^i is the estimate maintained by node l of the i class packets on the links downstream to node l and finally, D_1^i is the expected delay in node l itself. Let us apply equation (10), at nodes 2 and N-1 shown in Fig. 4.

$$\begin{aligned} d_2^i &= du_2^i + D_2^i + dd_2^i \\ d_{n-1}^i &= du_{n-1}^i + D_{n-1}^i + dd_{n-1}^i \end{aligned} \tag{11}$$

From equations (11) it is clear that as long as the packets are still near to their source nodes the upstream delay estimate, du_2^i , is not a very good approximation to d_2^i (ETE delay). However, as the packets get closer to the destination node, the upstream delay estimate becomes a good approximation to the ETE delay. That is

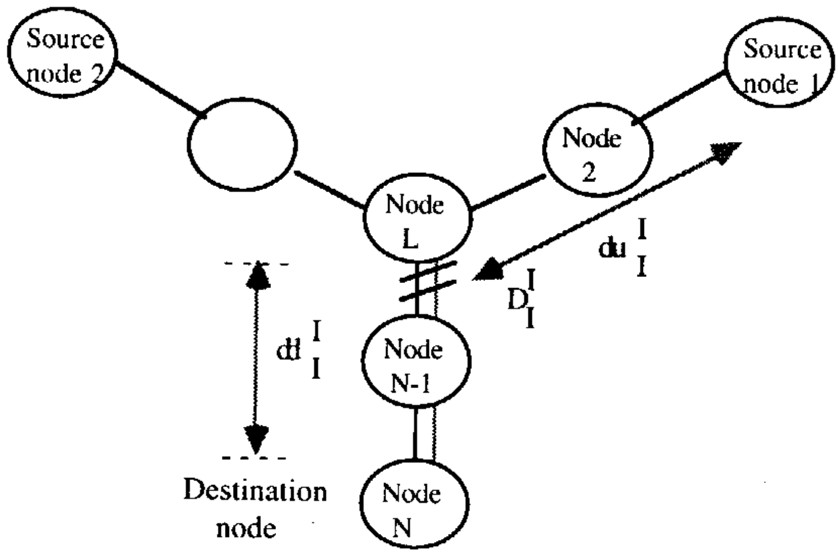

Fig. 4. A sample (route) path in a given packet network.

$du^{i}_{n-1} \cong d^{i}_{n-1}$. On the other hand, since the basic goal of the transmission scheduling scheme is to equalize the ETE delay experienced by all packet classes in the network. This has actually been achieved in our priority scheme as follows. Each node along a given path participates in the delay equalization process according to its estimation of the delay which has already been experienced by the packets along the path down to itself. Eventually, the delay equalization task will be, completely, achieved by the time packets arriving to their destinations.

At this point some explanations, related to the priority scheme which is being presented, should be made:

a) Why choosing the time-dependent priority scheme in particular? In fact, the point of departure for choosing the priority scheme is the conservation theorem [7], which states that for the Kleinrock's packet switching model with the Poisson assumption, the mean ETE delay is identical for all the work-conserving, non preemptive priority disciplines. Therefore, as long as the Poisson assumption gives accurate results, the selection of the scheduling discipline would not affect the mean ETE delay over all the traffic classes in the network.

b) One might want to use rather a simpler, fixed, priority scheme. The hopdependent discipline is a good example where the priority of a given packet is set equal to the number of hops (links) it spans. Results of this scheme have shown [12], a significant disparity of the ETE delay among different traffic classes. This was attributed to the discrete set of priority (number of hops) used thereof. Furthermore, a priority discipline with continuous parameters (such as the one being adopted), was recommended to be used instead in order to better manipulate the delays of various traffic classes.

c) Even with the time-dependent priority discipline. One might still want to use a simpler version of it where some (time-stamp) updating mechanism is used to insure that all the nodes in the network know the ETE delay, b_i , of each traffic class. It is not difficult to see that such an updating mechanism requires some sort of supervision mechanism for insuring that all the nodes receive their delay updating information concurrently. It is our belief that achieving such an important task will necessitate a centralized control of the network in order to provide the synchronization required for all the updating messages. Furthermore, implementing the (centralized/distributed) updating mechanism will impose a significant waste of the network bandwidth which is otherwise available for handling the actual data. What is left, at this point, is to shed some light on the implementation of the priority scheme we have developed and discuss some of the results obtained. This is the subject of the next Section.

5. Simulation Results

In this Section, we present results from simulation in order to demonstrate the performance of the (upstream) packet delay estimation algorithm and also to show the performance of the transmission scheduling mechanism where the delay estimation system is used in conjunction with the priority scheme presented in Section 4.

At first, the delay estimation algorithm was implemented in the subnetwork shown in Fig. 5. For our purpose, simulation of a whole network was not needed. Rather a sample path (route) in a network is what is required to be simulated. In the following, we add on the subnetwork model presented in Section 2.

In the simulation model all terminals are made to generate messages at the same rate with the time intervals between all consecutive messages drawn from an exponential

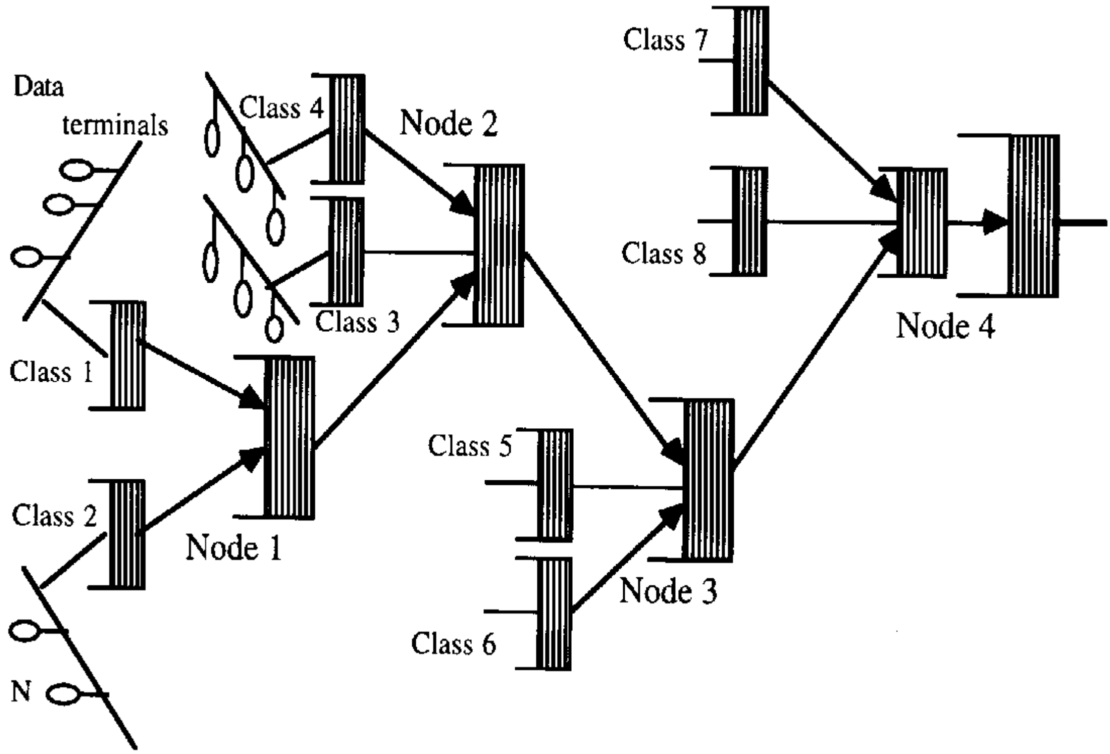

Fig. 5. Subnetwork simulation model.

distribution. The message lengths are broken down to an integer number of packets each of fixed length (16 m Sec.). The queues in both the nodes and terminals were introduced in order to improve the transmission line utilization since (user) messages are separated by a relatively long idle period. For simplicity of the simulation model, it is assumed that all packets are destined to the same destination node 4.

For each class i packets, after traversing a given link, we record its arrival time to the next node. This constitutes all the observations we need to compute (in an on-line manner) the upstream delay estimation. This has been achieved using equations (8),(9) in Subsection (3.2). Note that in the computation process the estimation algorithm quantizes the packet's interarrival times. At the beginning of the simulation course, we have chosen an arbitrary quantization step-size. The effect of the quantization step on the performance of the estimation process will be investigated later.

The first objective of the simulation is to verify that the estimation algorithm is capable of tracking the upstream delay experienced by packet classes which has traversed different number of hops in the subnetwork. Therefore, we have chosen to implement

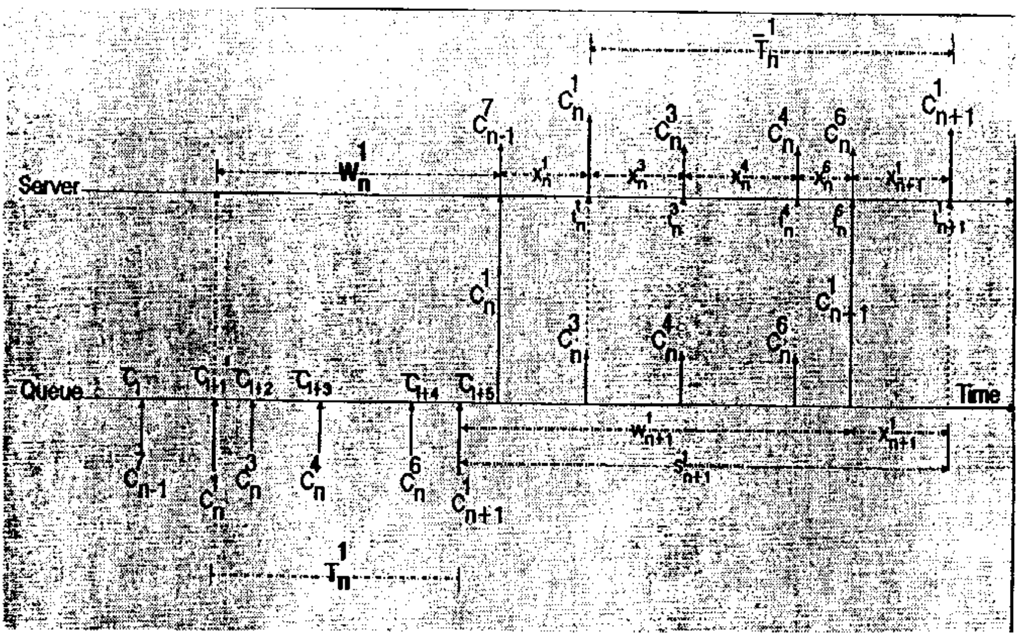

Fig. 6. Timing diagram.

the delay estimator at node 4 only. This has provided us with a delay estimation for packets has traversed only one link (i.e., classes 7, 8), two links (i.e., classes 5,6), three links (i.e., classes 3,4) and four links (i.e., classes 1,2). Figure 7 through Fig. 10, show the resultant delay estimations as a function of the simulation time. Note that the delay estimator is capable of tracking the level of the packet delays as well as its variations due to the variation in the traffic pattern offered to different nodes in the subnetwork. During this part of simulation, we have noticed that the resultant estimations are significantly sensitive to variations in the transition rate μ. This is illustrated in Fig.11. On the other hand, the delay estimation is less sensitive to variations in λ as can be seen from Fig. 12. In fact, this was expected since the transition rate μ depends on the state of the (modulating) intensity process. which is not the case for the transition rate λ.

Figure 13 shows the behavior of the estimation algorithm for different values of the (time) quantization step-sizes. As can be seen, when the step-size is small enough. This guarantees tracking of all the information imbeded in the packet's interarrival times. However. when the step-size is relatively large, the estimator tends to loose site of such an information.

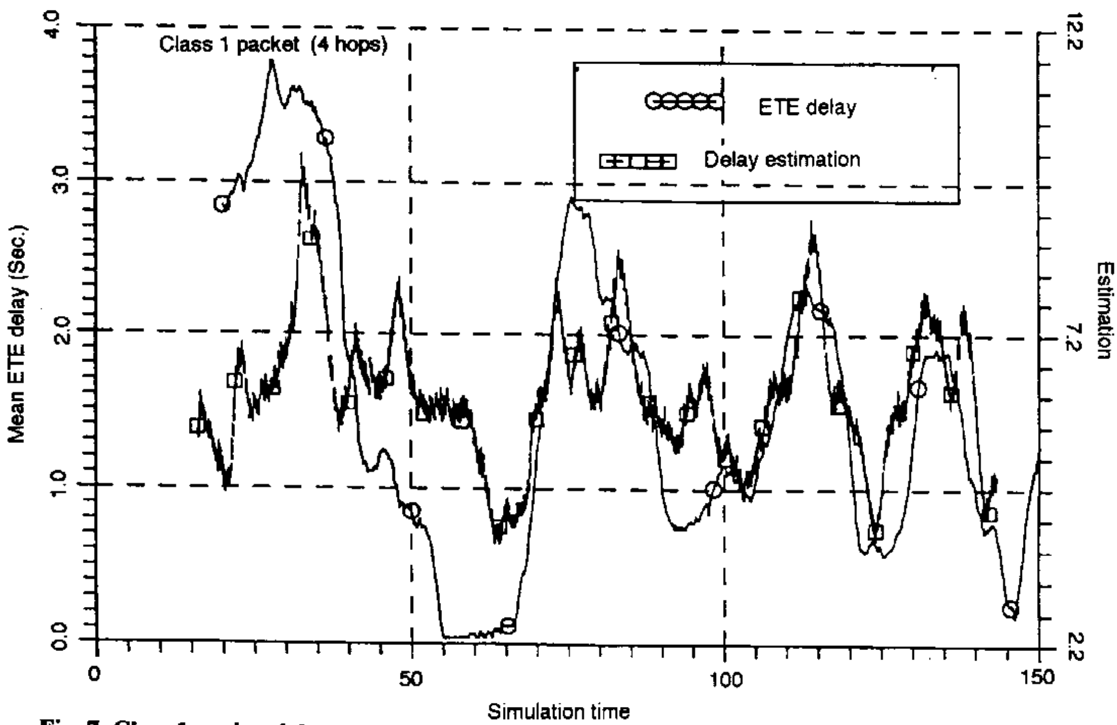

Fig. 7. Class 1 packet delay estimation.

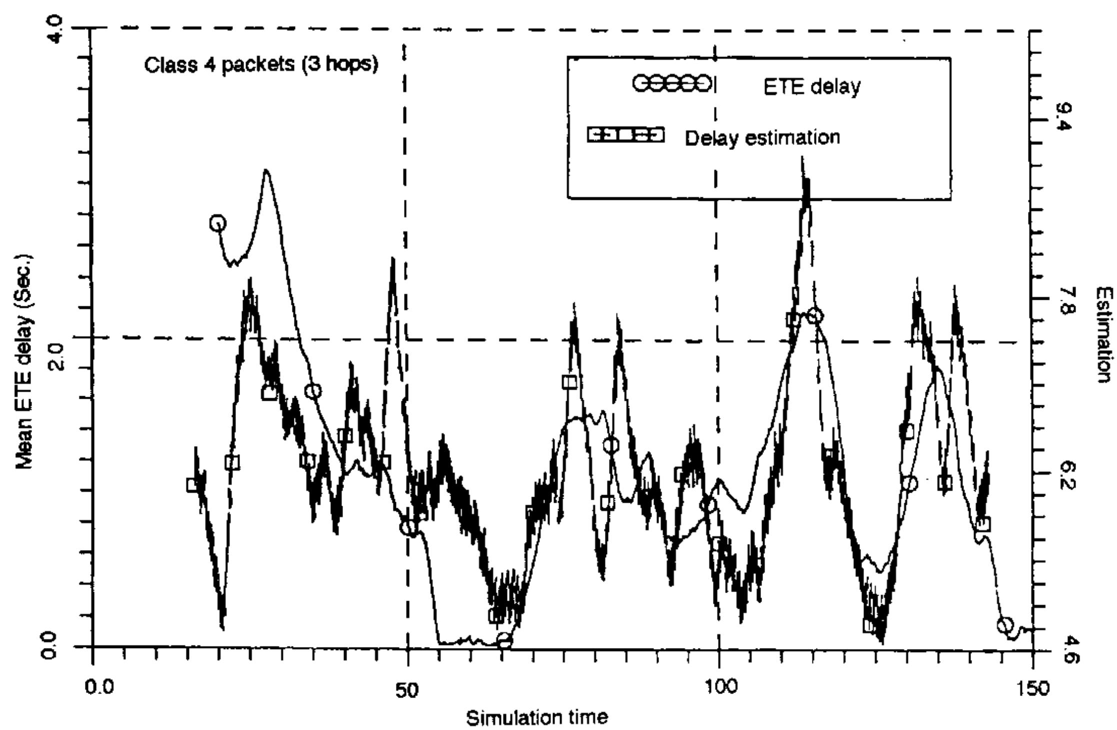

Fig. 8. Class 4 packets delay estimation.

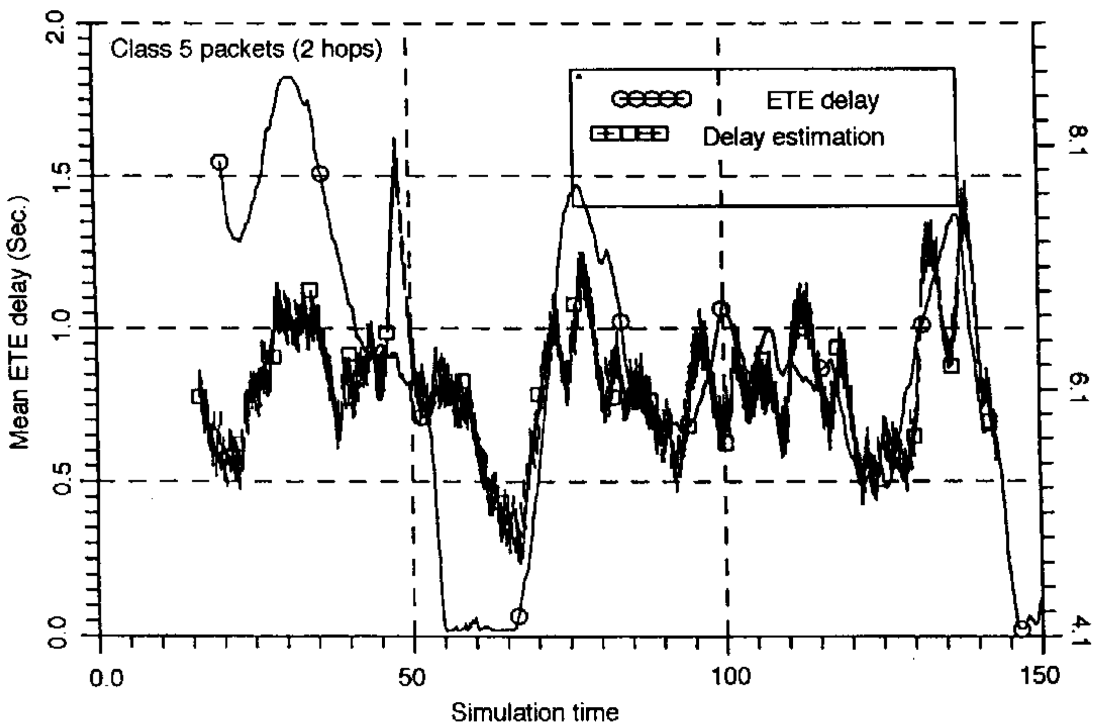

Fig. 9. Class 5 packets delay estimation.

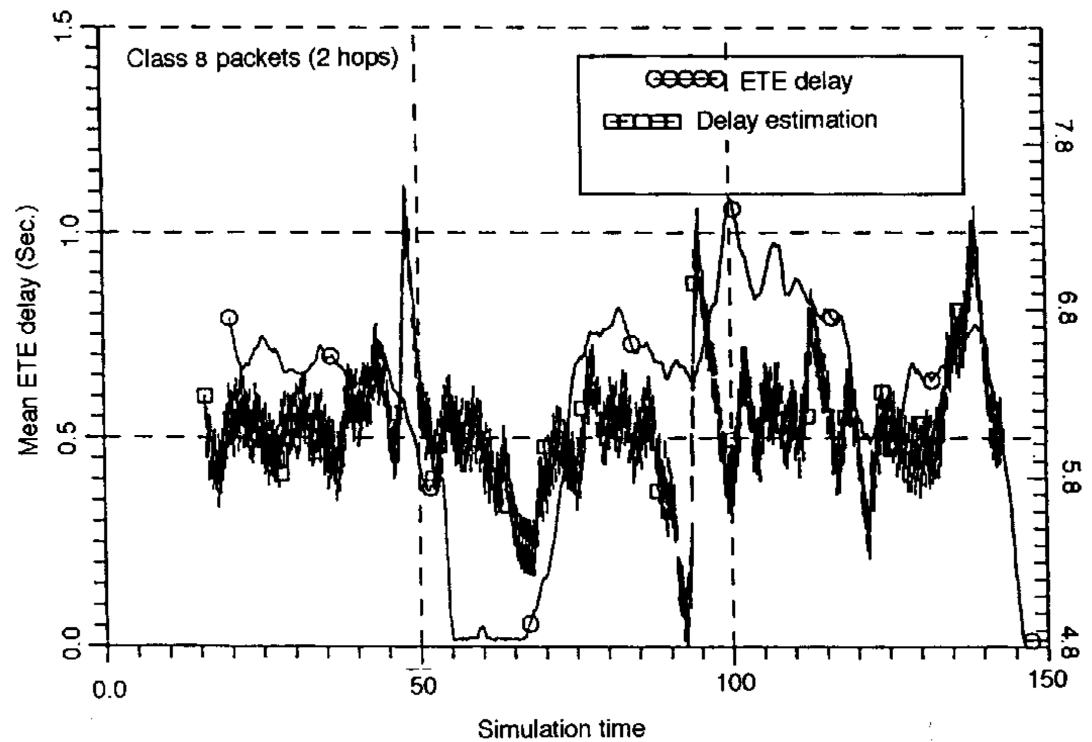

Fig. 10. Class 8 packets delay estimation.

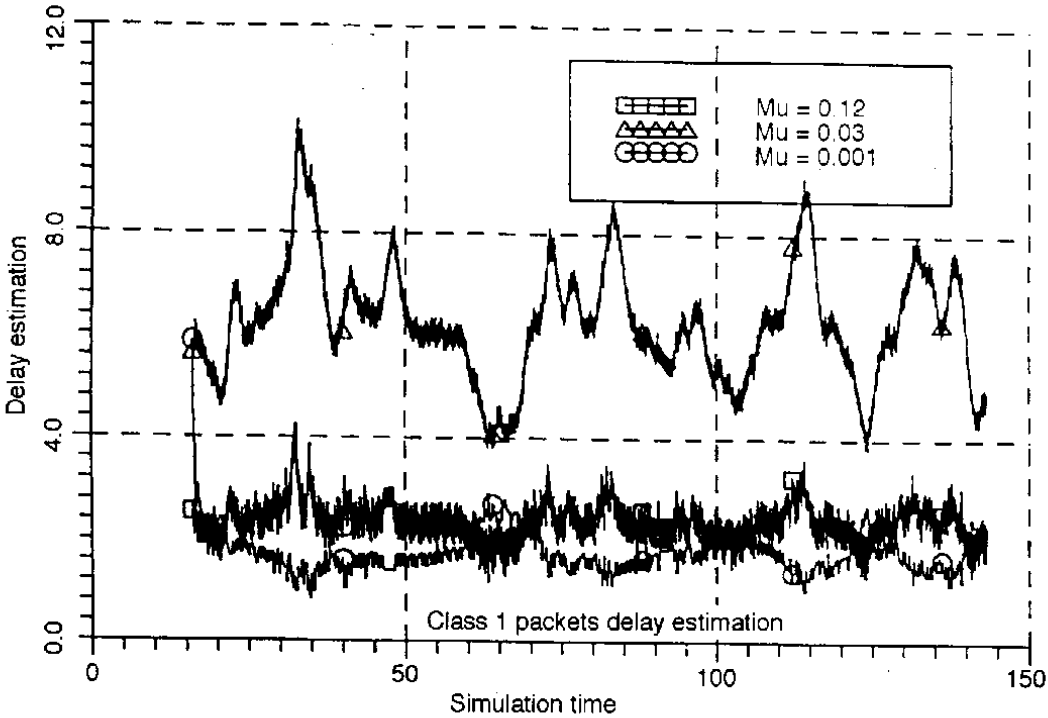

Fig. 11. Effect of 'Mu' on the delay estimation.

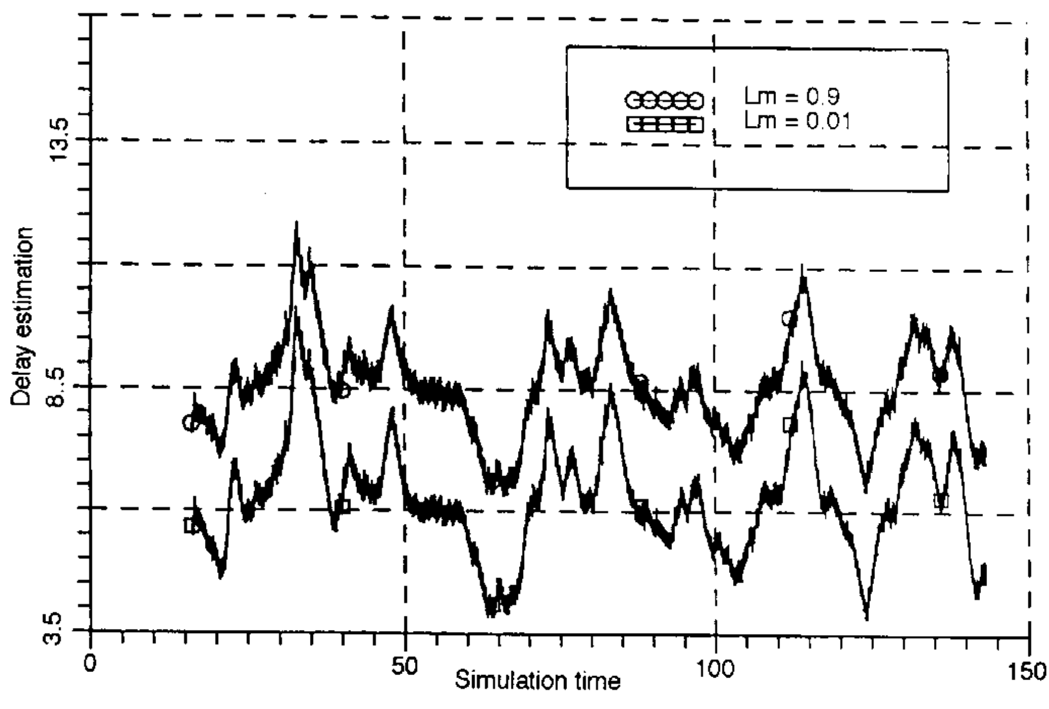

Fig. 12. Effect of 'Lm' ont he delay estimation.

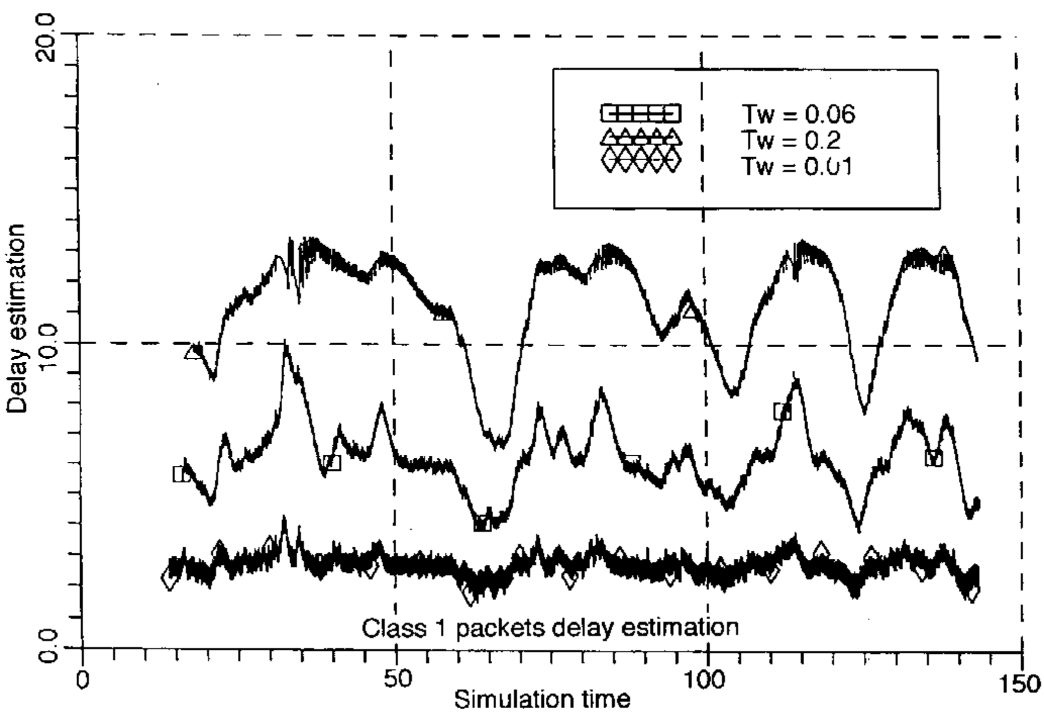

Fig. 13. Effect of 'Tw' on the delay estimation.

In the second part of the simulations carried out, the objective is to investigate the effect on the packet's ETE delays when the estimation algorithm is used in conjunction with the time-dependent priority scheme. We started by implementing the priority scheme in node 4 only. The resultant ETE delays are shown in Figs. 14 through 16. It is clear that node 4 has redistributed its own queuing delays among the different packet classes. This satisfies part of our objective.
What is left is to conduct a few more simulation runs with all nodes in the subnetwork exercising the same priority routine on their respective packet classes. By so doing, the following results which are depicted in Figs. 17 through 20, were obtained. As can be seen, the packets belonging to class 7 (for example), which happened to traverse less number of hops, in the subnetwork, are having their ETE delays increased significantly by some amount. The same amount of delays (work conserving) were reduced from the ETE delays of the other packet classes which have traversed more hops. This can easily be seen in Figs. 21 and 22. It is clear that the priority system has reduced the differences between the ETE delays for all the packet classes.

6. Summary and Conclusions

Addressed in this paper is the question of fairness in packetized computer data networks. We have asserted that equal mean ETE delay for all packet classes in the

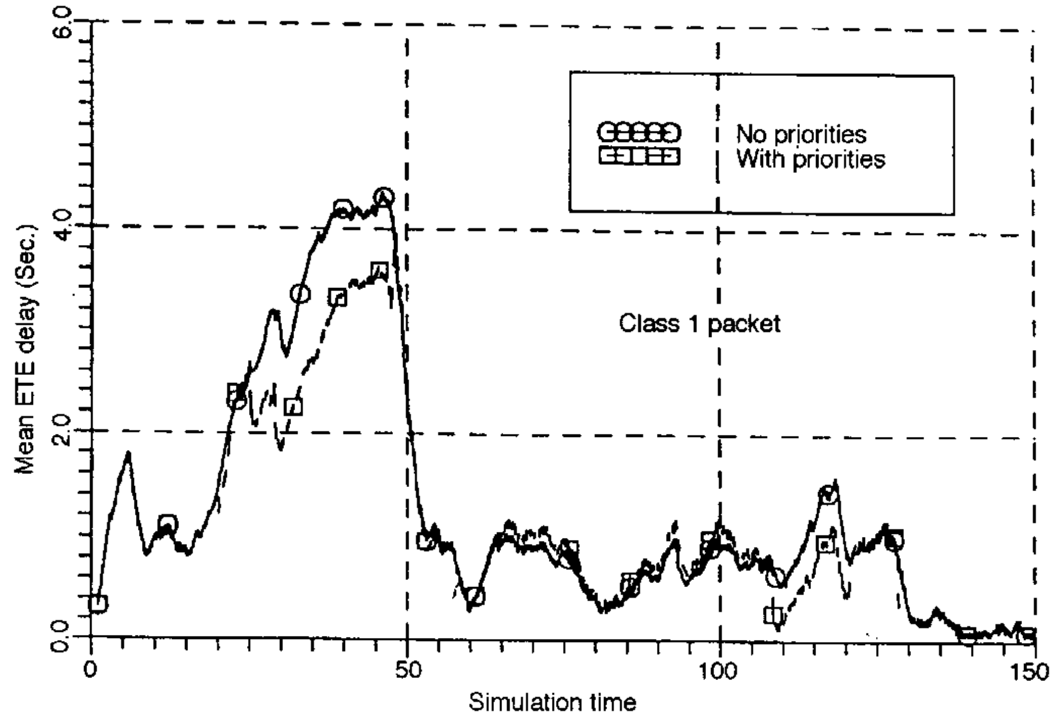

Fig. 14. Effect of priority on ETE delay.

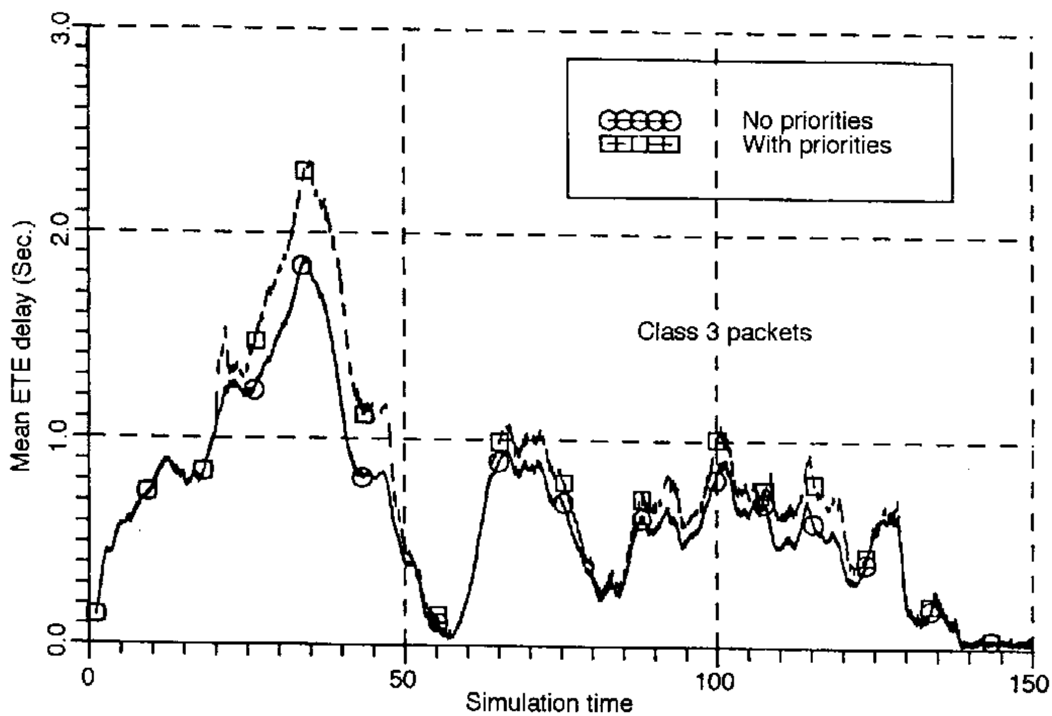

Fig. 15. Effect of priority on ETE delay.

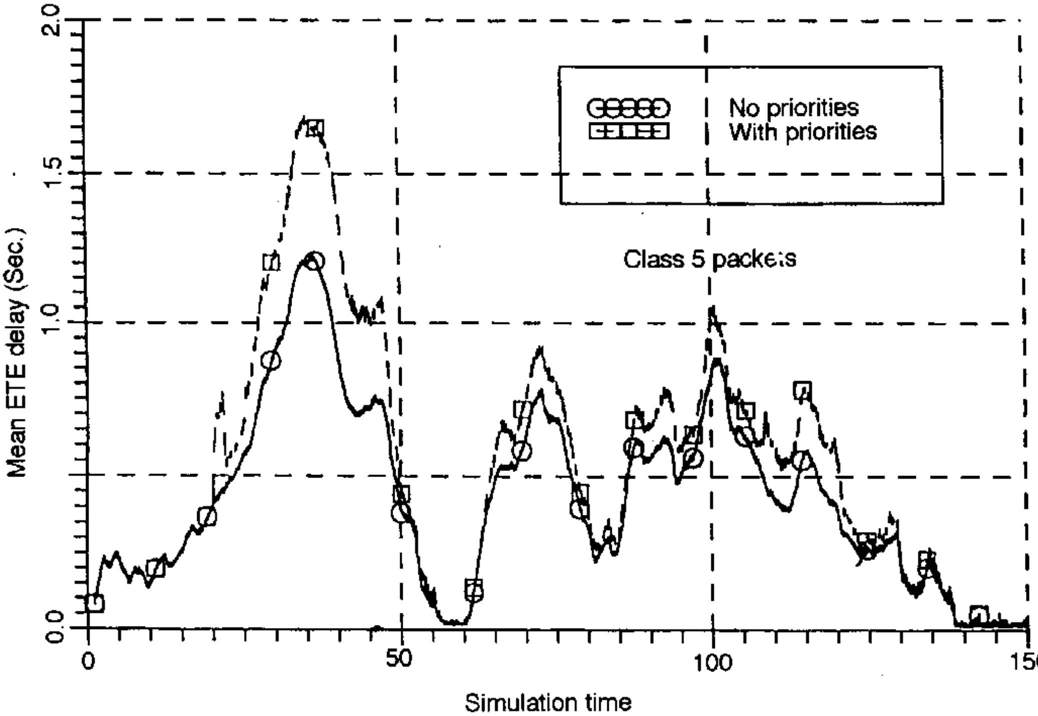

Fig. 16. Effect of priority on ETE delay.

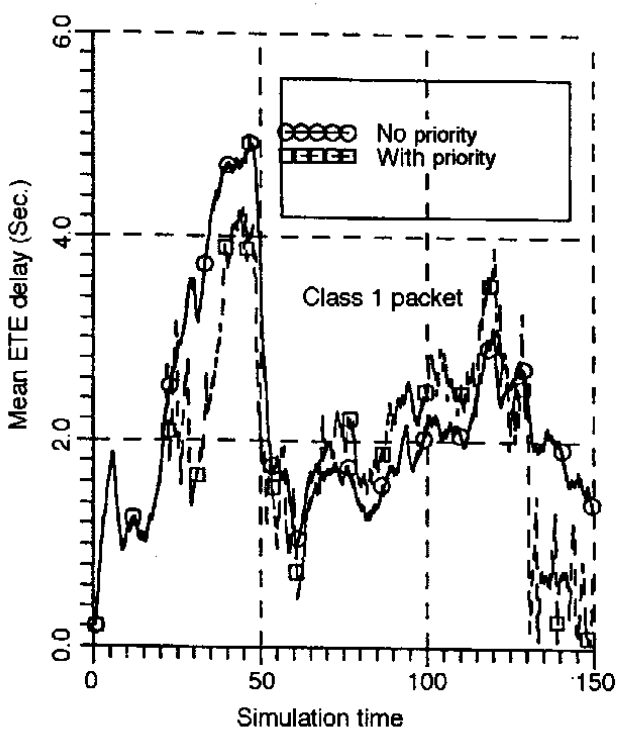

Fig. 17. Effect of priority.

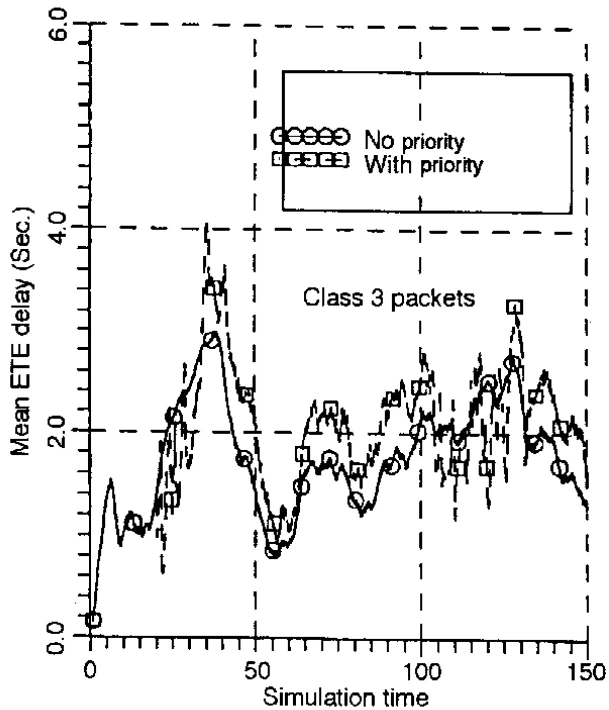

Fig. 18. Effect of priority.

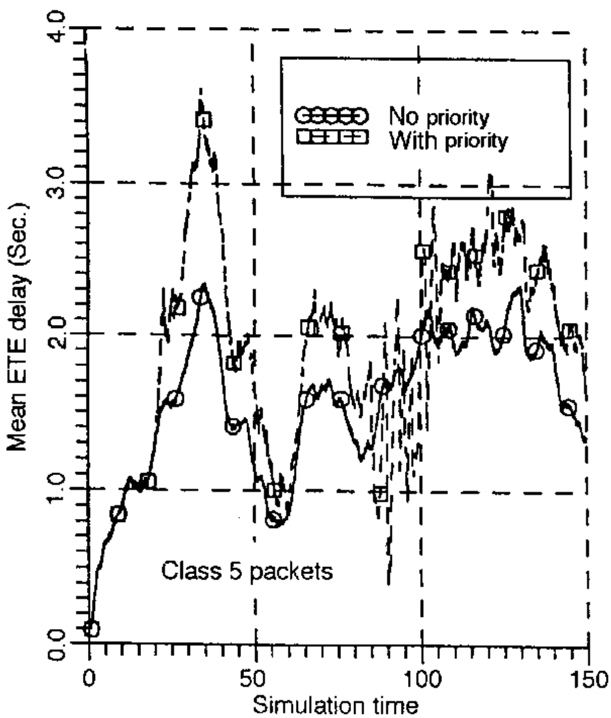

Fig. 19. Effect of priority.

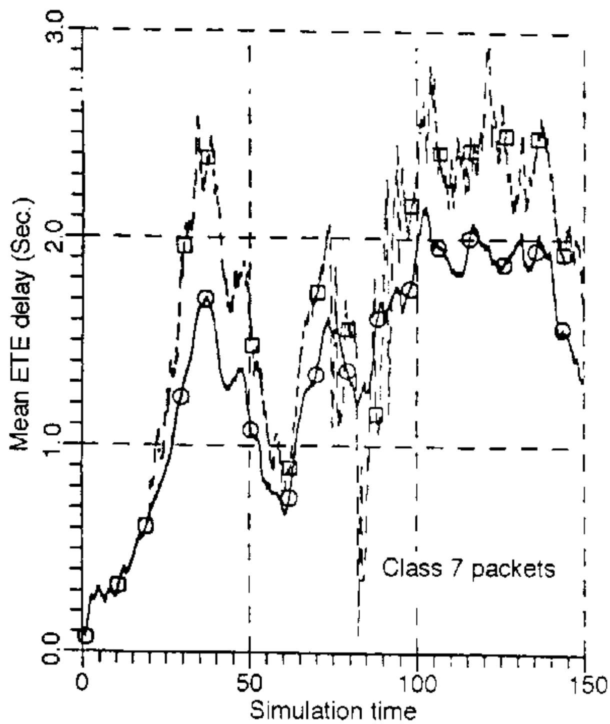

Fig. 20. Effect of priority.

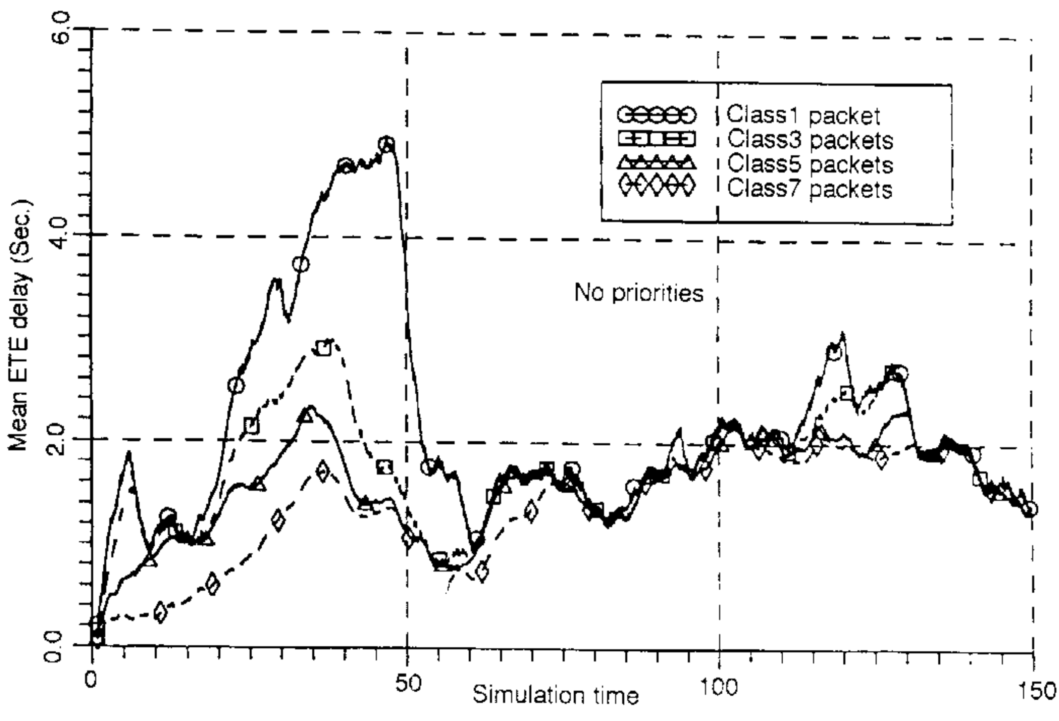

Fig. 21. Mean ETE delay with no priorities.

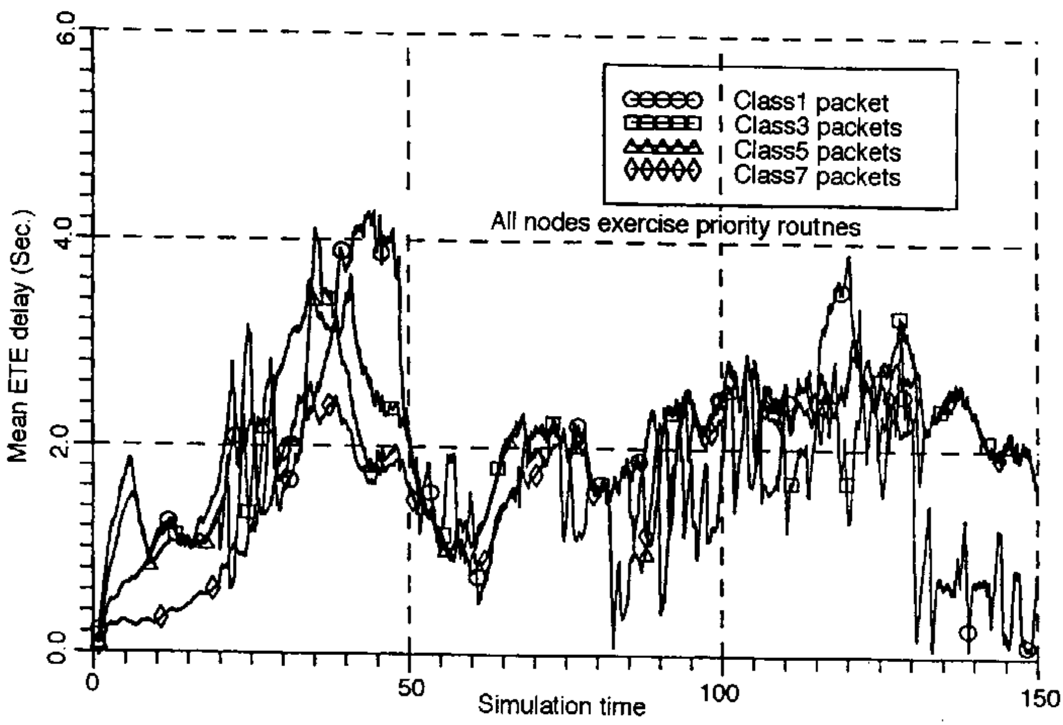

Fig. 22. Effect of priorities on overall delays.

network is a reasonable measure of fairness. In that respect, it is shown that transmission scheduling can improve fairness of the networks. Kleinrock's time-dependent priority discipline was considered due to its continuous parameters that one can use to manipulate the class delays.

In developing the present priority scheme, no knowledge of the state of the network delays is required to be exchanged between the involved parties in the network. Instead, each node in the network estimates the current state of the packet delays based on its observation of the packet arrival times. This approach is better suited for decentralized networks.

The paper presents a model for a Markov modulated point process and a scheme for recursive estimation of the network (upstream) delays. This is used in conjunction with the Kleinrock's priority system for real-time implementation of a dynamic priority system which is capable of equalizing the mean ETE delays of different traffic classes in the network. Implementation of this system is considered to be relatively simple, and the processing overhead is relatively small.

Simulation results investigating the performance of the delay estimation scheme and the adaptivity of the dynamic priority system are included. We conjectured and supported by the simulation results that by properly adjusting the (delay) estimator parameters, the resultant dynamic priority scheme can improve the (ETE delay) fairness in packet data networks.

An important observation concerning the delay estimation algorithm is its sensitivity to the values of μ. Thus, several different values of μ were tried. We expect that, because of this sensitivity to μ, it will be possible to use an adaptive procedure to estimate it in an actual implementation of the dynamic priority scheme.

APPENDIX A

We present here a brief summary of the results on the modeling of discrete-time point processes following Segall [23].

Consider the sequence of observations $\{n(t)\}_{t=1}^{\infty}$ with $n(t) = 0$ or $n(t) = 1$ being the only possibilities for each t. Suppose the probability that $n(t) = 1$ is influenced by previous occurrences as well as by some other related sequence $\{x(t)\}_{t=1}^{\infty}$. The factors that may affect the occurrence probability at time t are the past observations denoted

$$n^{t-1} = \{n(1), n(2), \ldots, x(t)\}$$

and the past and present of the related sequence

$$x^{t} = \{x(1), x(2), \ldots, x(t)\}$$

The information carried by these signals is denoted by the σ-algebra generated by them,

$$\beta_{t-1} = \sigma\{n^{t-1}, x^{t}\} .$$

We then define a (.) by

$$P_r\{n(t) = 1/\beta_{t-1}\} = 1 - P_r\{n(t) = 0/\beta_{\tau-1}\}$$
$$\cong a(t, n^{t-1}, x^{t})$$

Then

$$E^{\beta_{t-1}}\{n(t)\} \cong E\{n(t)/\beta_{t-1}\} = a(t, n^{t-1}, x^t), \text{ where}$$

$$E^{\beta_{t-1}}\{z\} \cong E\{z/\beta_{t-1}\} \text{ is the conditional expectation of z given } \beta_{t-1}.$$

If we define

$$w(t) \equiv n(t) - a(t, n^{t-1}, x^t) \text{ then}$$

$$E^{\beta_{t-1}}\{w(t)\} = 0,$$

which says, roughly, that w(t) is unpredictable given the information represented by β_{t-1}. Similarly, if we write

$$x(t+1) = E^{\beta_{t-1}}\{x(t+1)\} + x(t+1) - E^{\beta_{t-1}}\{x(t+1)\}$$

and define

$$f(t, n^{t-1}, x^t) = E^{\beta_{t-1}}\{x(t+1)\}$$

$$u(t) = x(t+1) - E^{\beta_{t-1}}\{x(t+1)\}$$

we obtain

$$E^{\beta_{t-1}}\{u(t)\} = 0$$

Assembling all of the above gives

$$x(t+1) = f(t, n^{t-1}, x^t) + u(t)$$

$$n(t) = a(t, n^{t-1}, x^t) + w(t)$$

These two equations simply reflect the fact that any observation sequence can be divided into the sum of a predictable part and an unpredictable part.

References

[1] Schwartz, M. and Stern, T.E. "Routing Techniques Used in Computer Communication Networks." *IEEE Trans. Commun.*, COM-28, (Apr. 1980), 539-552.

[2] McQuillan, M. "Routing Algorithms for Computer Networks - A Survey." In : *Proc. 1977 Nat. Telecomm. Conf.* (Dec. 1977).

[3] Reiser. "A Queuing Network Analysis of Computer Communication Networks with Window Flow Control." *IEEE Trans. Commun.*, COM-27, (Aug. 1979), 1199-1209.

[4] Gerla and Kleinrock, L."Flow Control: A Comparative Survey." *IEEE Trans. Commun.*, COM-28, (Apr. 1980), 553-574.

[5] Irland, "Buffer Management in a Packet Switch." *IEEE Trans. Commun.*, COM-26, (Mar. 1978), 328- 337.

[6] Kamoun and Kleinrock, L. "Analysis of Shared Finite Storage in a Computer Network Environment Under General Traffic Conditions." *IEEE Trans. Commun.* COM-28, (1980), 992-1003.

[7] Kleinrock, L. *Queuing Systems: Volume II, Computer Applications.* New York: Wiley Interscience,1976.

[8] Gerla. *The Design of Store-and-forward Networks for Computer Communications.*" School Eng. Appl. Sc., Univ. California, Los Angeles, Rep. UCLA-ENG-7319 (Jan. 1973).

[9] Kumar, B. and Jaffe, J. "A New Approach to Performance Oriented Flow Control." *IEEE Trans. Commun.*, COM-29, (Apr. 1981), 427-435.

[10] Gerla and Staskausakas, M. "Fairness in Flow Controlled Networks." In: *Proc. Int. Conf. Commun.* (1981), 63. 2. 1- 63. 2. 5.

[11] Thaker, H. and Cain, J. B. "Interactions Between Routing and Flow Control Algorithms." *IEEE Trans. Commun.*, COM-34, (1986), 269-277.

[12] Wong, W.; Sauve, J.P., and Field, J.A."A Study of Fairness in Packet Switching Networks," *IEEE Trans. Commun.*, COM-30, (1982), 346-353.

[13] Davis, H. A. "The Representation of Martingales of Jump Processes." *SIAM J. Contr. Optimiz.*, 14, (Apr. 1976), 623-638.

[14] Segall; Davis, M. H. A., and Kailath, T. "Nonlinear Filtering with Counting Observations." *IEEE Trans. Inform. Theory*, IT-21, (Mar. 1975), 53-58.

[15] Bremaud. "A Martingale Approach to Point Processes," Univ. Calif., Berkeley, Memo ERL M345, (Aug. 1972).

[16] Segall. "Stochastic Process in Estimation Theory." *IEEE Trans. Inform. Theory*,IT-22, (May 1976), 275-285.

[17] Kleinrock, L. *Communication Nets, Stochastic Message Flow and Delay.* New York: McGraw-Hill, 1964, out of print (reprint by Dover, 1972).

[18] Rudin, H. "Buffered Packet Switching: A Queue with Clustered Arrivals." *Proc. International Switching Symposium*, Boston, 1972.

[19] Lewis, P. A. W. " A Branching Poisson Process Model for the Analysis of Computer

Failure Patterns." *J. R. Stat. Soc.* 26, (1964).

[20] Heffes, H. A. "Class of Data Traffic Processes Covariance Function Characterization and Related Queuing Results." *Bell Syst. Tech. J.*, 59, 897-929.

[21] Burman, D. Y. and Smith, D. R. "Asymptotic Analysis of a Queuing Model with Bursty Traffic." *Bell Syst. Tech. J.*, 621, (1983), 1433-1452.

[22] Weinstein, C. J. "Fractional Speech Loss and Talker Activity Model for TASI and for Packet-switched Speech." *IEEE Trans. Commun*,COM-26, No. 8, (Aug. 1978), 1253-1257.

[23] Segall. "Recursive Estimation from Discrete Time Point Processes." *IEEE Trans. Inform. Theory*, IT-22, (July 1976), 422-431.

نحو معادلة التأخير في شبكات حزم المعلومات باستخدام تقدير زمن التأخير

جمال عبدالفضيل محمد

قسم الاتصالات، كلية الهندسة والتقنية، جامعة حلوان،

حلوان - مصر

ملخص البحث . يتناول هذا البحث أسلوبًا جديدًا لمعادلة تأخير حزم المعلومات عند مرورها في شبكات حزم المعلومات. تعتمد الطريقة الجديدة على نظام لتقدير قيم تأخير الحزم عند مرورها في شبكات المعلومات. ولقد أثبتت نتائج هذا البحث فاعلية واضحة وقدرة على تحقيق التساوي في تأخيرات حزم المعلومات المختلفة رغم اختلاف أطوال مساراتها. الأمر الذي يحقق مبدأ المساواة في مستوى الأداء بين غالبية المشتركين رغم تفاوت مواقعهم الجغرافية على شبكة حزم المعلومات.