The Utilization of Priorities on Token Ring Networks

Jeffery H. Peden and Alfred C. Weaver

Computer Science Report No. TR-89-19
August 15, 1989

The Utilization of Priorities on Token Ring Networks

Jeffery H. Peden and Alfred C. Weaver
Department of Computer Science
Thornton Hall
University of Virginia
Charlottesville, Virginia 22903
(804) 979-7529

Abstract — Many local area networks supply the user with multiple packet priority levels for network access. The intent is to discriminate in favor of those messages which the sender deems most important. This discrimination between higher and lower priority messages is realized as a difference in their end-to-end delivery delays, with higher priority messages having less delay than lower priority messages.

There are, however, problems associated with priority schemes. Among these are the overhead generated by prioritizing messages, and the determination of which messages belong in each priority level. If the correct decisions are not made, optimal network performance will not result.

Our thesis is that, in the normal environment where the network is not overloaded, *no* packet priority levels should be implemented (i.e., exactly one message class should be used); furthermore, using multiple packet priorities in this environment will generally degrade overall performance. For priority operation to be meaningful, priority service must be given at the point of the service bottleneck. Our experience is that the bottleneck is not network access, but rather the software processing above network access (e.g., OSI protocols). If a message has not been treated expeditiously prior to its bid for network access, assigning a high network access priority does little to decrease its latency.

I. INTRODUCTION

Many local area network access protocols provide the user with message priorities in an attempt to ensure that more important messages experience less delay than less important messages. Priority mechanisms fall into two main categories. The first is the use of timers to control the relative amount of time given to each priority (''timed token'' protocols), and the second is to use a priority reservation scheme whereby stations on the network negotiate the lowest priority at which messages may currently be transmitted.

There are four main problems concerning the use of priorities, however. These problems are (1) the overhead associated with their use, (2) the fact that higher priority messages do not (and cannot) always preempt lower priority messages, (3) determining the number of priorities to be used in a given installation, and (4) determining the fraction of available bandwidth to be assigned to each priority class.

A. Timer Mechanisms

Two well-know local area networks which use timer-controlled priority mechanisms are the IEEE 802.4 Token Bus [1] and the Fiber Distributed Data Interface (FDDI) [2]. The Token Bus has four priority levels; FDDI has no protocol-defined upper limit on the number of priorities. The basic operation of this type of priority mechanism is as follows.

Using a negotiation process which is detailed in their respective protocol definitions, these networks decide on a synchronous token hold time and target token rotation times for each of the lower priority classes. At token reception, any messages in the synchronous priority class are serviced either until none are left, or until the synchronous token hold time has been exceeded. At this point, the station computes the difference between the target rotation time and actual token rotation time for the next highest priority class, and transmits messages of that class until the computed amount of time has been exceeded or no messages of that class remain (if the computed difference is less than zero, this priority class is not served). This process iterates within a station until all priority classes have been handled appropriately.

B. Reservation Mechanisms

Two well-known local area networks using a reservation priority mechanism are the IEEE 802.5 Token Ring [3] and the SAE High Speed Ring Bus (HSRB) [4]. The 802.5 token ring allows eight priority levels; the HSRB allows eight *asynchronous* priority levels and one *synchronous* priority. The basic operation of the reservation mechanism for both is described as follows.

Stations which have enqueued packets attempt to reserve the priority of the rotation token. This ensures that the token is transmitted at the highest priority of any enqueued packet on the ring (with the exception of higher priority packets which arrive after the reservation bid has been made). When a station receives the token, only those packets whose priority is equal to or higher than that of the token may be transmitted. Transmission of these packets continues until either all appropriate packets have been transmitted, or until the MAC calculates that the token holding timer would expire before the completion of another packet transmission (in the case of the SAE token ring, only a single packet per received token may be transmitted). The transmitting station then emits the token at the highest requested priority.

The two protocols differ somewhat in the methods used to lower the priority of the token. In the case of the 802.5 token ring, if a station raises the priority of the token from X to Y, and later receives a token of priority Y, with no reservations higher than X, it returns the priority of the token to X. The SAE protocol states that the new token is always transmitted at a priority equal to the received reservation (the reservation is always initialized to zero by the sending station in both protocols). Note that this will have the effect of lowering the token's priority approximately one token cycle sooner than would happen on the 802.5 token ring (we are assuming here that the short message protocol of the SAE token ring is not used, since it ignores priority).

II. PERFORMANCE MODELS

It is beyond the scope of this paper to deal with complete analytic models for priority mechanisms. However, in this section we present some analytic results which deal with the overhead associated with the handling of priorities. Our models here deal only with the overhead associated with MAC layer protocols. For more general treatments of models for local area networks, the reader is referred to Schwartz [5], Konheim and Meister [6], Bux [7], and Kuehn [8].

A. Timed Token Model

Priority mechanisms implemented via the use of timers allow the network implementor to determine the amount of service each priority level will receive. That is, the network implementor can set the timers for each priority such that each priority will receive service up to some specified throughput level. The intent is to determine the token cycle time at which the specified throughput level is reached, thus obtaining the desired timer value. The model presented here is taken from Gorur and Weaver [9].

1. Maximum Throughput Levels

Defining α to be the throughput level (as a fraction of network capacity) at which service for a certain priority shall cease, we have

$$\alpha = \frac{N \cdot \Lambda_\alpha \cdot l_M}{C \cdot \Gamma_\alpha} \qquad (1)$$

where

$N \equiv$ the number of network stations

$\Lambda_\alpha \equiv$ the number of packets per second transmitted at each station when network throughput is α

$l_M \equiv$ the mean packet length

$C \equiv$ the network transmission rate in bits per second

$\Gamma_\alpha \equiv$ the mean token cycle time when network throughput is α

Solving for Λ_α,

$$\Lambda_\alpha = \frac{\alpha \cdot C \cdot \Gamma_\alpha}{N \cdot l_M} \tag{2}$$

Since the mean number of packets arriving is equal to the mean number of packets transmitted,

$$\Lambda_\alpha = \Gamma_\alpha \sum_{i=1}^{n} \lambda_i \tag{3}$$

where

$\lambda_i \equiv$ the arrival rate in packets per second of priority i

$n \equiv$ the number of priority classes in use

Note that there is an infinite number of priority class arrival rates that satisfy the above equation. It is possible to express the token cycle time Γ_α as

$$\Gamma_\alpha = N\ (X_T + X_M \cdot \Lambda_\alpha) \tag{4}$$

where

$X_T \equiv$ the token transmission and propagation time

$X_M \equiv$ packet transmission and propagation time

Performing appropriate substitutions, we obtain the equation for the value of the token cycle time beyond which throughput for priority k ceases, which is the same as the timer setting τ_k for this priority class:

$$\tau_k = \frac{N \cdot X_T}{1 - N \cdot X_M \cdot \sum_{i=k}^{n} \lambda_i}, \quad n > 1, \;\; k < n \tag{5}$$

2. *Limiting Throughput*

It is evident that this priority mechanism cannot prevent the transmission of a low priority packet at one station before the transmission of a higher priority packet at the next station, except when the maximum throughput level of the lower priority has been reached (and then only probabilistically). Thus the priority mechanism is inherently inconsistent, that is, the strict ordering of priority levels is only maintained (for the most part) when high load is present.

Since priority performance is controlled by the throughput level of that priority (and hence the load on the network), unless the network designer is extremely careful when setting the timer values to control throughput levels, optimum performance will not result, since maximum throughput values for all priority levels can be significantly reduced by incorrect timer settings. For example, if the network designer sets the timer value for the lowest priority such that its throughput cuts off too soon, the maximum throughput of the network as a whole will be degraded unnecessarily, since the higher priority levels do not automatically increase their load to raise throughput levels to the maximum possible.

B. *Reservation Model*

1. *Reservation Overhead*

The number of packets that may be transmitted upon token reception has a large effect on the discrimination between the various priority classes in a reservation scheme. In the HSRB, only one packet may be transmitted per token reception; in the case of the 802.5 token ring, the maximum number of packets which may be transmitted per received token is controlled by the token holding timer. In the case of the 802.5 token ring, the maximum number of packets ξ transmitted per token reception is given by

$$\xi = \lfloor \Omega \mu \rfloor \tag{6}$$

where

$\Omega \equiv$ the setting of the token holding timer in seconds

$\mu \equiv$ the maximum service rate in packets per second

The *floor* function is necessary because the 802.5 standard states that the token holding timer may not be overrun; that is, prior to each packet transmission, the token holding timer is checked to see if there is enough time remaining to complete the pending transmission.

The number of attempted priority reservations R made in one token cycle (assuming Poisson arrivals) is given by

$$R = N(1 - \exp(-\Lambda(E[V]\xi^{-\Phi(N-1)} + \frac{1}{\mu}))) \tag{7}$$

where

$N \equiv$ the number of ring stations

$\Lambda \equiv$ the sum of the arrival rates of packets of all priorities at a single station

$\Phi \equiv$ the total load contribution of a single station, $\Phi = \frac{\Lambda}{\mu}$

$E[V] \equiv$ the mean token vacation time, i.e., the time between a station's transmission of a free token and that station's reception of the next free token

The number of attempted reservations may include multiple requests for the same priority; however, multiple requests for the same priority are counted as a single request for the purposes of priority overhead. The number of distinct requests is the total number of different token priorities requested during the token cycle. We now define $\mathbf{X}$ to be the vector random variable which takes on the value of the total number of reservation requests made, regardless of whether or not they were distinct, and we also define $\underline{\mathbf{X}}$ to be the total number of X_i components having non-zero values.

The probability distribution of **X** is given by

$$P(\mathbf{X}=\mathbf{k}) = \begin{pmatrix} R+1 \\ k_1, k_2, \cdots, k_n \end{pmatrix} \prod_{i=1}^{n} \left[\frac{r_i}{n} \right]^{k_i} \tag{8}$$

where

$r_i \equiv$ the proportion of the total load of packets of priority i

$n \equiv$ the number of priority classes being offered to the network

The reason for the term $R+1$ rather than R is that the priority of the token counts as a reservation for the purposes of computing the number of distinct reservation requests, since a reservation request for a priority equal to that of the token will have no effect.

Using the above definitions, the equation for the total delay overhead caused by the reservation mechanism is given by

$$\sum_{\underline{\mathbf{X}}}^{R} (\underline{\mathbf{X}} \cdot P(\mathbf{X}=\mathbf{k}) \cdot (E[V] + \frac{1}{\mu})) \tag{9}$$

2. *Effective Load*

The minimum possible load that can affect the delay of priority i packets is the load contribution of priority i plus the load contribution of all higher priorities. The maximum possible load affecting priority i packets is the total network load. The *actual* load affecting priority i packets lies between these limits, and is called the *effective* load. The equation for the effective load σ_i for priority i is

$$\sigma_i = \Phi - \Phi \xi^{-\Phi(N-1)} (\Phi - \delta_i) \tag{10}$$

where

$\delta_i \equiv$ the load contribution at a single station of all priorities greater than or equal to i

The effective load for priority i decreases in proportion to the load at a station, and increases in proportion to the number of stations and the maximum number of packets transmitted per received token. This is the reason that the added delay introduced by the reservation mechanism affects *all* priorities, not just those priorities which are lower than the token priority.

Figures 1 - 4 show the effective load for the various priorities as functions of offered load, single or multiple packet per token service, the number of stations, and the maximum number of packets transmitted per token received. Figure 1 shows the effective load for a configuration of 3 stations and single packet per token (SPPT) service (it is shown in [10] that a priority system has more impact as the number of stations becomes small). Figure 2 shows the effective load for the same configuration with the exception that the number of packets per token is set to 39, which is the default (in an 802.5 token ring) for this packet length (32 octets). Figure 3 shows the effective load for a configuration of 3 stations and 0.95 offered load, while varying the maximum number of packets per token. Figure 4 shows the effective load for a configuration of 0.95 offered load and SPPT service, while varying the number of stations. It can be seen from these four graphs that only under certain circumstances (high offered load, short packets, short token hold times, and a small number of network stations) does the effective load for the highest priority drop *significantly* below the effective load for the lowest priority. This indicates that the priority scheme does not generally reduce the effective load for the higher priorities enough to compensate for the added overhead.

III. PERFORMANCE MEASURES

In this section we present results from simulations we have made of the IEEE 802.5 token ring and the Fiber Data Distributed Interface token ring. These results reinforce our thesis that priority usage should be restricted to those situations where it is absolutely necessary; otherwise degraded network performance can result.

A. FDDI

In this section we examine some of the effects of threshold timer values on an FDDI token ring [11]. Fig. 5 shows delay for all eight asynchronous priorities plus the single synchronous priority on a network configuration of 100 active stations, 10 kilometer ring circumference, and 128 octet packets. All priority threshold values are set to the target token rotation time, which in this configuration is 9.0 msec. The difference between the maximum and minimum delays is only a few microseconds, which is not significant. The mean delay for all priorities at the maximum offered load shown is approximately 2.3 msec.

Fig. 6 shows delays for an identical configuration, except that priority timer threshold values have been varied from 3.0 msec for priority 0 (lowest) to 6.5 msec for priority 7 (the highest asynchronous priority) with a step value of 0.5 msec. The two effects noted are the shut-off of service for priority 0 with a consequent reduction of overall delay for the other priorities. We shall see shortly, however, that this drop in delay is accompanied by a reduction in overall throughput. The delay for the upper priorities at the highest offered load shown is approximately 1.6 msec, which is a reduction of only 0.7 msec from the previous graph.

Fig. 7 again reports delay for the same configuration, except that priority threshold values have been lowered further. In this graph, the thresholds are varied from 1.0 msec for priority 0 (lowest) to 4.5 msec for priority 7 (highest), again with a step value of 0.5 msec. Here service shutoff for priority 0 occurs at about 0.7 offered load, with shutoff for priority 1 at about 0.85

offered load. The mean delay for the other priority levels is approximately 0.9 msec; this reduction in delay is again accompanied by a loss of overall throughput.

Fig. 8 shows throughput levels for the three previous graphs. The maximum throughput for the configuration in Fig. 5 is 0.92, that for Fig. 6 is 0.89, and maximum throughput for Fig. 7 is 0.80. Therefore, the drop in delay from 2.3 msec to 0.9 msec in Fig. 5 was accompanied by a drop in throughput from 0.92 to 0.80. At the 100 megabits/sec data rate of the FDDI token ring, this is a loss of approximately 12 megabits/sec, which we consider to be an unacceptable penalty.

B. 802.5 Token Ring

Here we present some simulation results showing the impact of the priority scheme on an 802.5 token ring. Fig. 9 shows a network configuration consisting of 40 active stations, 32 octet packets, and 1 megabit/sec transmission speed. The upper curves show the delays for all eight priorities allowed by the 802.5 protocol. For this simulation, we divided the load evenly among all eight priority levels. Note that all the delays are significantly above the delay shown when the entire load is offered at a single priority.

Fig. 10 shows the delays for all eight priority levels in a network identical to the one above, with the exception that there are only three active stations. There is now some noticeable difference between the delays for the highest and lowest priorities, but it is still less than a factor of two. The delay for this configuration when the entire load is offered at a single priority is very close to that of the delay of the highest priority (approximately 2.2 milliseconds as opposed to about 1.95 milliseconds for the highest priority level).

It is evident that only in extreme cases does the priority mechanism decrease delay for high priority packets without adding unwanted delay to lower priority packets. It was shown in [10] that for the reservation priority mechanism used by the IEEE 802.5 token ring, optimal priority performance only occurs when the network has very few stations, packets are short, load is high, and token hold times are small.

IV. ASSIGNMENT OF PRIORITIES

We have shown that the use of network access priorities can have an adverse impact on the performance of a network. However, when implemented correctly, the use of priority levels can reduce the delay of high priority packets with only an insignificant addition to the delay of lower priority packets (that is, only occasionally will the lower priority packets be forced to wait for one of a higher priority).

The three most important factors to keep in mind are that the number of priority levels used should be as few as possible (we have never identified a case where more than four priority levels were necessary); most of the network load should be offered at the *normal* priority; and that a packet should be assigned its priority at the soonest possible moment after it has been submitted for transmission (e.g., before its submission to the application layer in the OSI reference model, rather than upon its arrival at the MAC sublayer). We discuss each of these factors in turn.

A. *Priority Levels*

We have shown that the overhead generated by priority use, whether in a reservation scheme or a timer scheme, can completely subsume any decrease in delay seen by lower priorities, as well as reduce total network throughput. This effect occurs mainly when there is much competition for the various priority levels among packets otherwise ready for transmission. If, however, the number of priority levels is reduced, this automatically reduces the overhead generated by packets desiring access to the various priority levels.

B. *Expected Queue Length*

It is generally unnecessary to introduce the overhead of network access priorities because network access priorities do not discriminate among enqueued packets unless the network is very nearly overloaded. To see why this is so, consider a modern FDDI network with 100 stations, each station sending 200 packets/second, with 600 bytes/packet. Assume the ring latency is

4,000 bit times. At the 100 Mbps transmission speed of FDDI, each station is contributing an offered load of 960,000 bits/second ($\rho = 0.0096$), so the total network offered load $N\rho = 0.96$. By any standard, an offered load of 96% of total capacity represents a very highly loaded network.

The expected queue length (measured at token arrival) in such a system is given by

$$E(Q) = \frac{\lambda\gamma(1-\rho)}{1-N\rho}$$

$$= \frac{(200)(0.00004)(0.9904)}{0.04}$$

$$= 0.19808$$

In other words, even in an FDDI network with an offered load of 0.96, a station will receive five tokens for every packet transmitted. Therefore, since packets are transmitted faster than they are enqueued, all packets are effectively carried at the same priority.

C. Effective Priorities

If network access priorities are generally ineffective, then where should the priority concept be implemented? To be effective, priority must be assigned early (i.e., as a message is submitted to the communication protocols) and the message must be handled expeditiously at every stage of its processing. Our experience with network performance measurement and evaluation in [12] and [13] shows conclusively that the bottleneck is not the network, but the protocol software ''above'' network access.

For instance, the OSI transport protocol [14] provides two message types (priorities), *normal* and *expedited*. The definition of *expedited* is that a message, once enqueued, must be delivered no later than any message enqueued thereafter — in other words, FIFO service, which is hardly a discriminating priority scheme. If the internationally standard protocols which are expected to feed data to the network only support two message classes, what is the value of eight (in 802.5) or an unlimited number (in FDDI) of network access priorities at the medium access control sublayer?

D. Synchronous Traffic

A legitimate concern with high-performance networks is how to handle synchronous traffic such as real-time voice. We contend that the solution to this problem is not the use of priorities, but rather the careful allocation of sufficient bandwidth by the network designer. As our results have shown, if sufficient bandwidth is not available for a particular application, then offered loads will be high and the use of a priority scheme may serve only to further degrade performance.

V. CONCLUSIONS

It is evident that unless very high loads are expected in the ordinary course of events, the use of priority mechanisms can severely degrade overall network performance. The use of a reservation priority mechanism can more than offset the reduced delay seen by the highest priority level if this mechanism is used in network configurations which do not warrant it. We have also seen that the incorrect use of a timer priority mechanism can severely impact the total network throughput.

The two mechanisms studied here have their own strengths and weaknesses. The use of a reservation mechanism is quite simple in that the network designer need not be concerned with tuning threshold values to obtain optimum performance. Another advantage of the reservation mechanism is that if all load is offered at a single priority, no added delay is experienced; that is, the mechanism is ''free'' if unused. However, since the network cannot tune the priority operation for optimal performance, it is unlikely that it will result.

The main advantage of timer threshold priority mechanisms is that the timer values can be tuned for optimal network performance. However, the process of determining optimal timer values requires information that is often unavailable *a priori* (e.g., traffic patterns) and may require many iterations before good network performance is achieved; it is also difficult to achieve optimal settings for dynamic network traffic. Another drawback to this method is that, if unused, this mechanism may still have an impact on delay and throughput since the timer values are always used by the MAC layer; if set too low, degraded throughput and delay could result even when only a single priority level is being used.

We have also shown that the *number* and relative loading of priority levels has an impact on delay. This indicates that the number of priority levels used should be no more than four (no more than two in most cases), with most network traffic being offered at a single priority level. We therefore believe that MAC layer priorities should be used as infrequently as possible.

REFERENCES

[1] *Token-Passing Bus Access Method and Physical Layer Specifications*, The Institute of Electrical and Electronics Engineers, Inc., 1985.

[2] *FDDI Token Ring Media Access Control Standard*, American National Standards Institute, draft proposed Standard X3T9.5/84-49, rev. 10, Feb. 1986.

[3] *Token Ring Access Method and Physical Layer Specifications*, The Institute of Electrical and Electronics Engineers, Inc., 1985.

[4] *SAE AS4074.2 High Speed Ring Bus*, Society of Automotive Engineers, final draft Standard, June 1987.

[5] M. Schwartz, *Telecommunications Networks: Protocols, Modeling and Analysis*, Addison-Wesley, 1987.

[6] A. G. Konheim and B. Meister, "Waiting Lines and Times in a System with Polling," *J. ACM 21*, 3 (July 1974), 470-490.

[7] W. Bux, "Local-Area Subnetworks: A Performance Comparison," *IEEE Transactions on Communications COM-29*, 10 (Oct. 1981) 1465-1473.

[8] P. J. Kuehn, "Multiqueue Systems with Nonexhaustive Cyclic Service," *Bell System Technical Journal 58*, 3 (Mar. 1979), 671-698.

[9] M. Gorur and A. C. Weaver, "Setting Target_Rotation_Times in an 802.4 Network," *IEEE Transactions on Industrial Electronics IE-35*, 3 (Aug. 1988).

[10] J. H. Peden and A. C. Weaver, "Are Priorities Useful in an 802.5 Token Ring?" *IEEE Transactions on Industrial Electronics IE-35*, 3 (Aug. 1988).

[11] R. A. Simonson, *Performance Analysis of the FDDI Token Ring*, Technical Report No. TR-88-02, Department of Computer Science, University of Virginia, Feb. 1988.

[12] W. T. Strayer and A. C. Weaver, "Performance Measurement of Data Transfer Services in MAP", *IEEE Networks*, May 1988.

[13] W. T. Strayer and A. C. Weaver, "Performance Measurements of Motorola's Implementation of MAP," Technical Report No. TR-88-10, Department of Computer Science, University of Virginia, April 1988.

[14] ISO 8072, "Information processing systems — Open Systems Interconnection — Transport Service Definition," June 15, 1986.

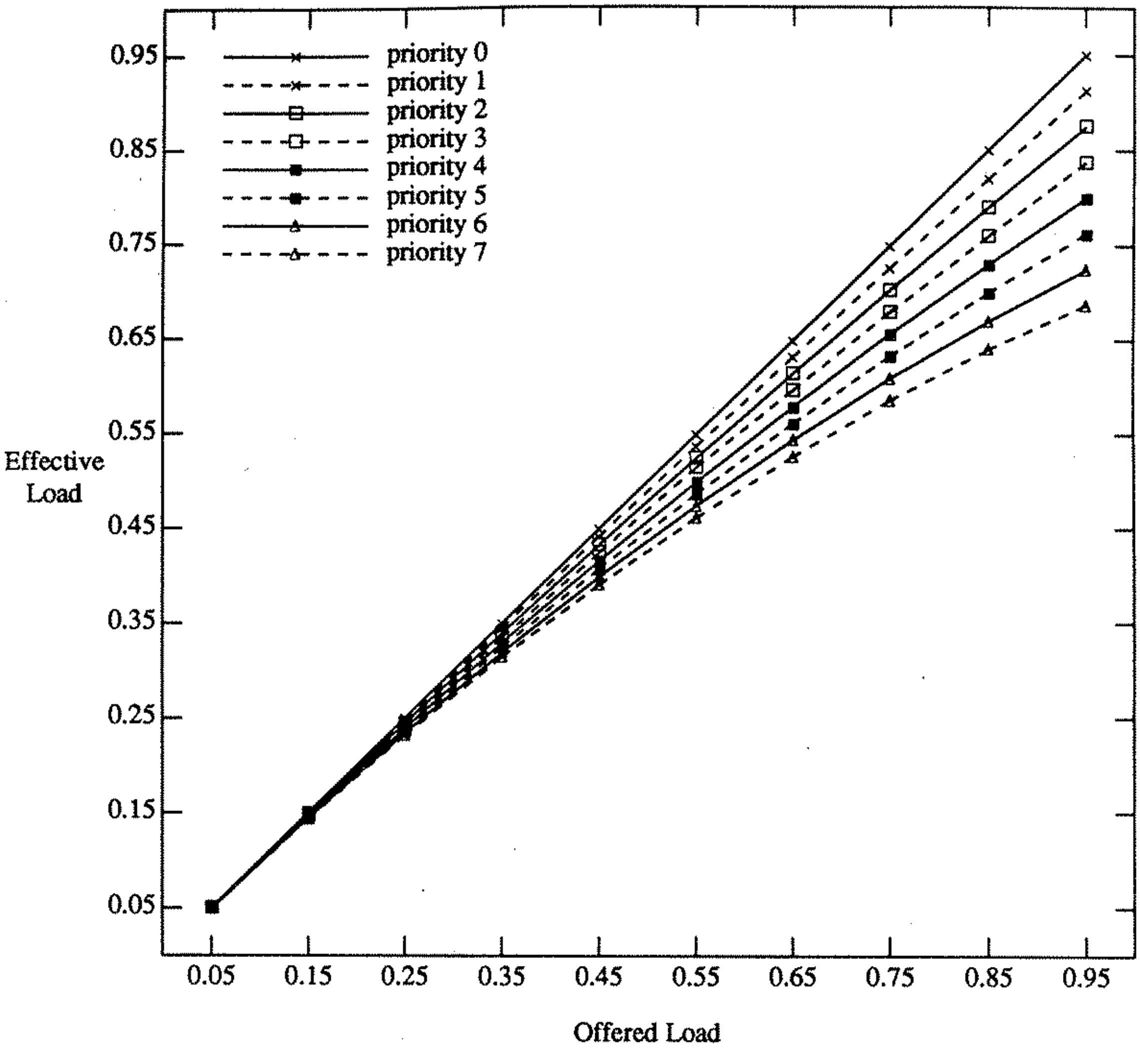

Figure 1.
Effective load, 3 stations and SPPT service.

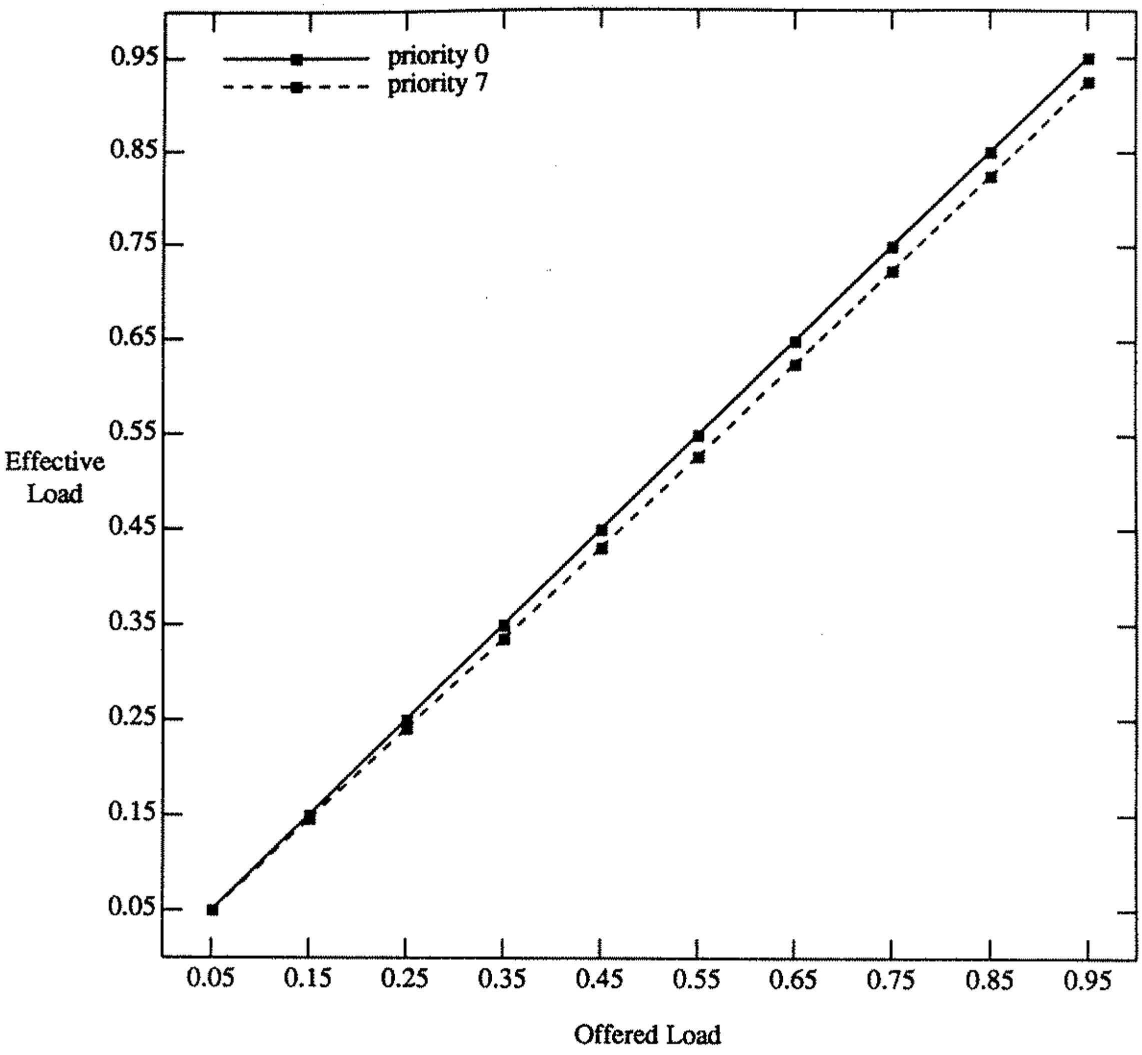

Figure 2.
Effective load, 3 stations, 39 packets per token (default).

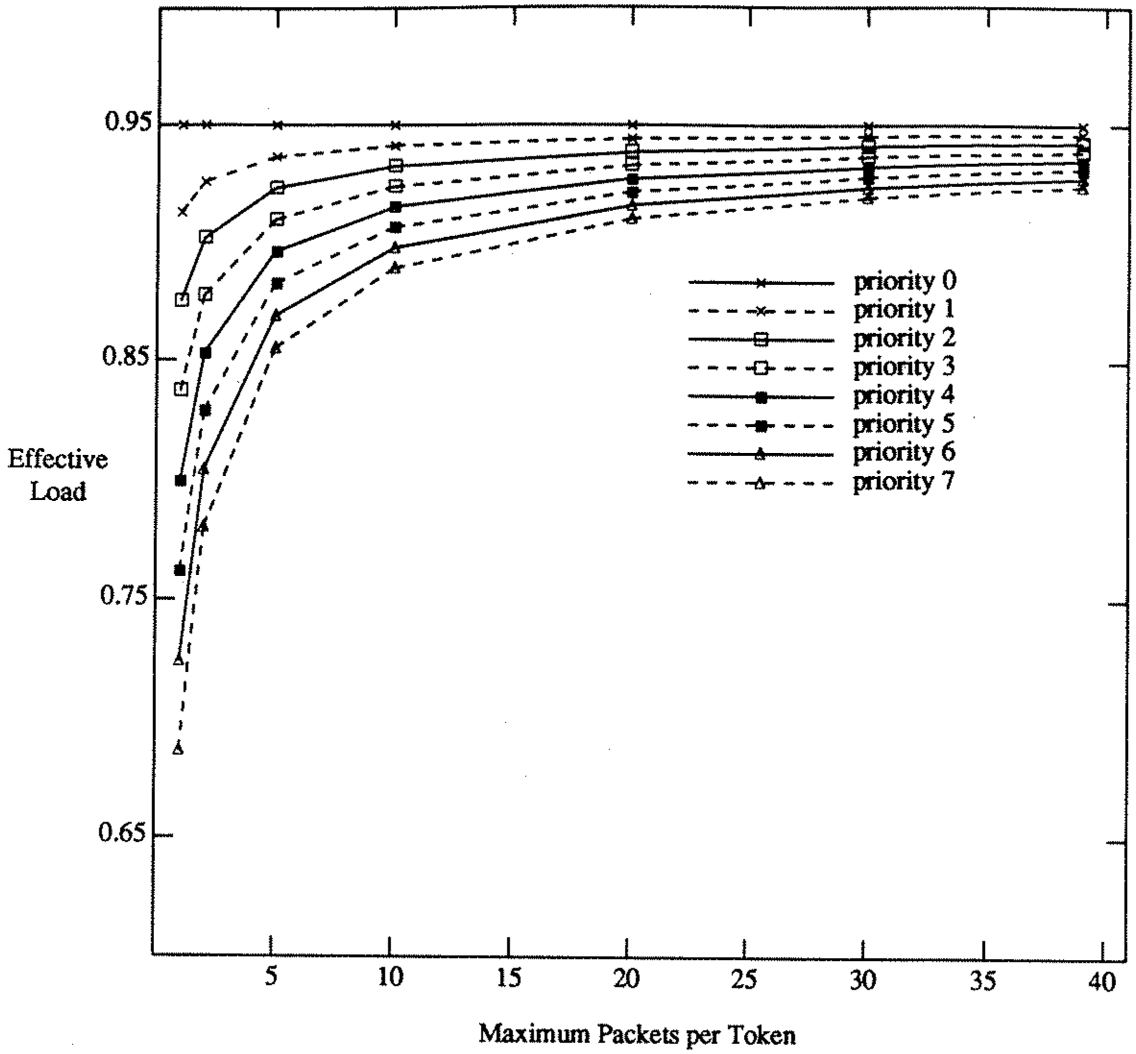

Figure 3.
Effective load, 3 stations, 0.95 offered load.

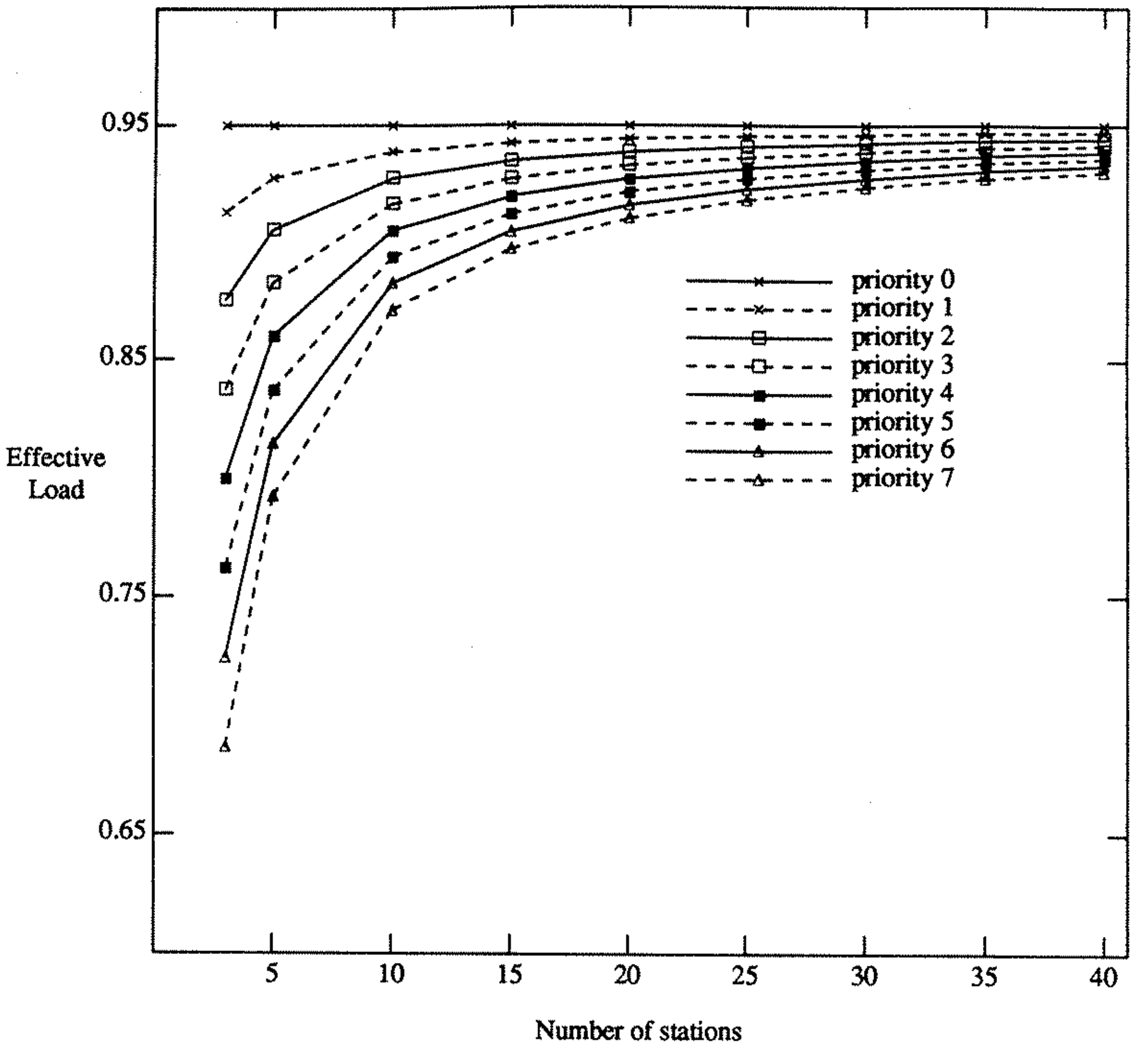

Figure 4.
Effective load, 0.95 offered load and SPPT service.

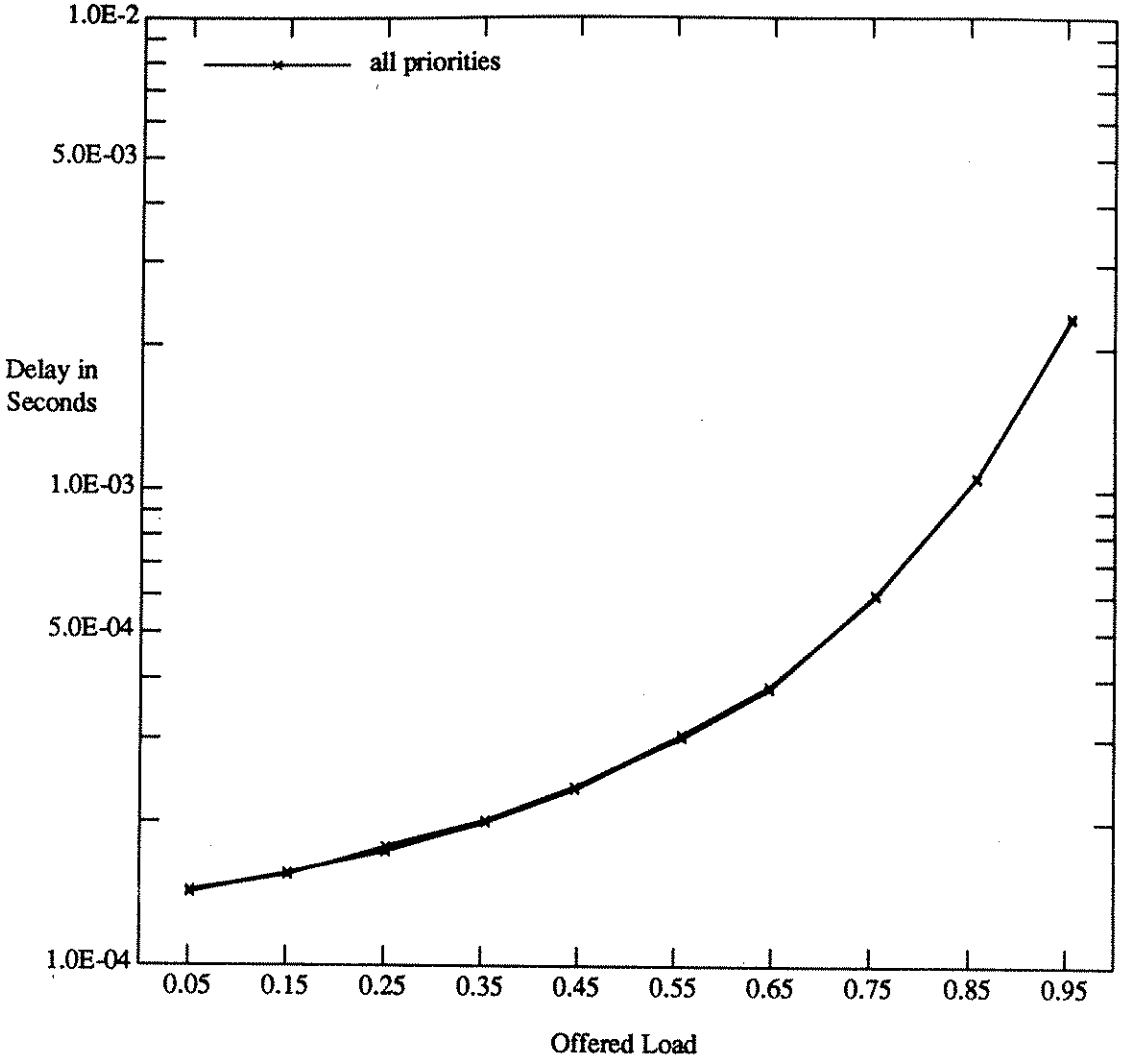

Figure 5.
Service delay for all priorities.

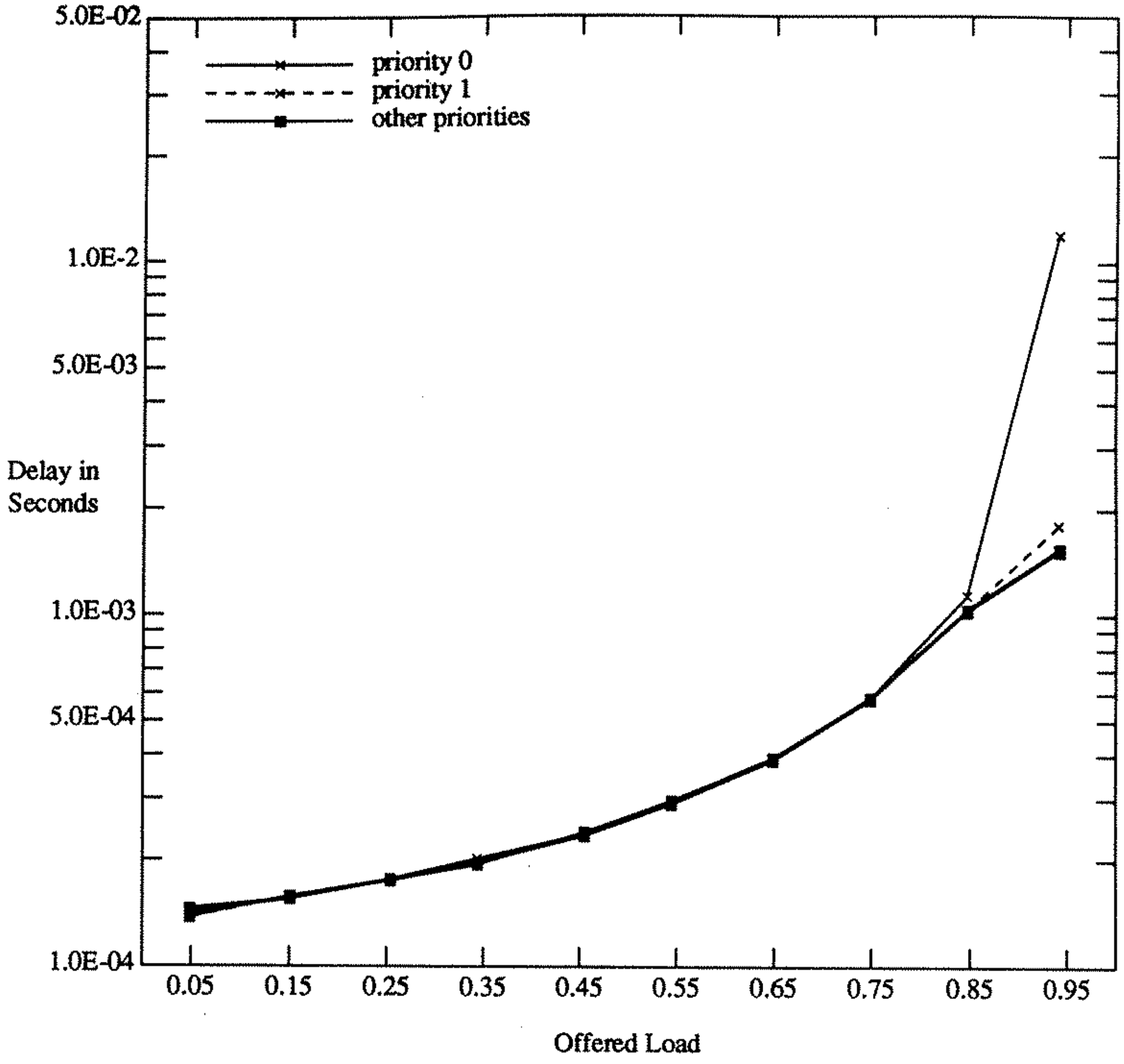

Figure 6.
Service delay, priority thresholds varied
from 3.0 ms to 6.5 ms by 0.5 ms.

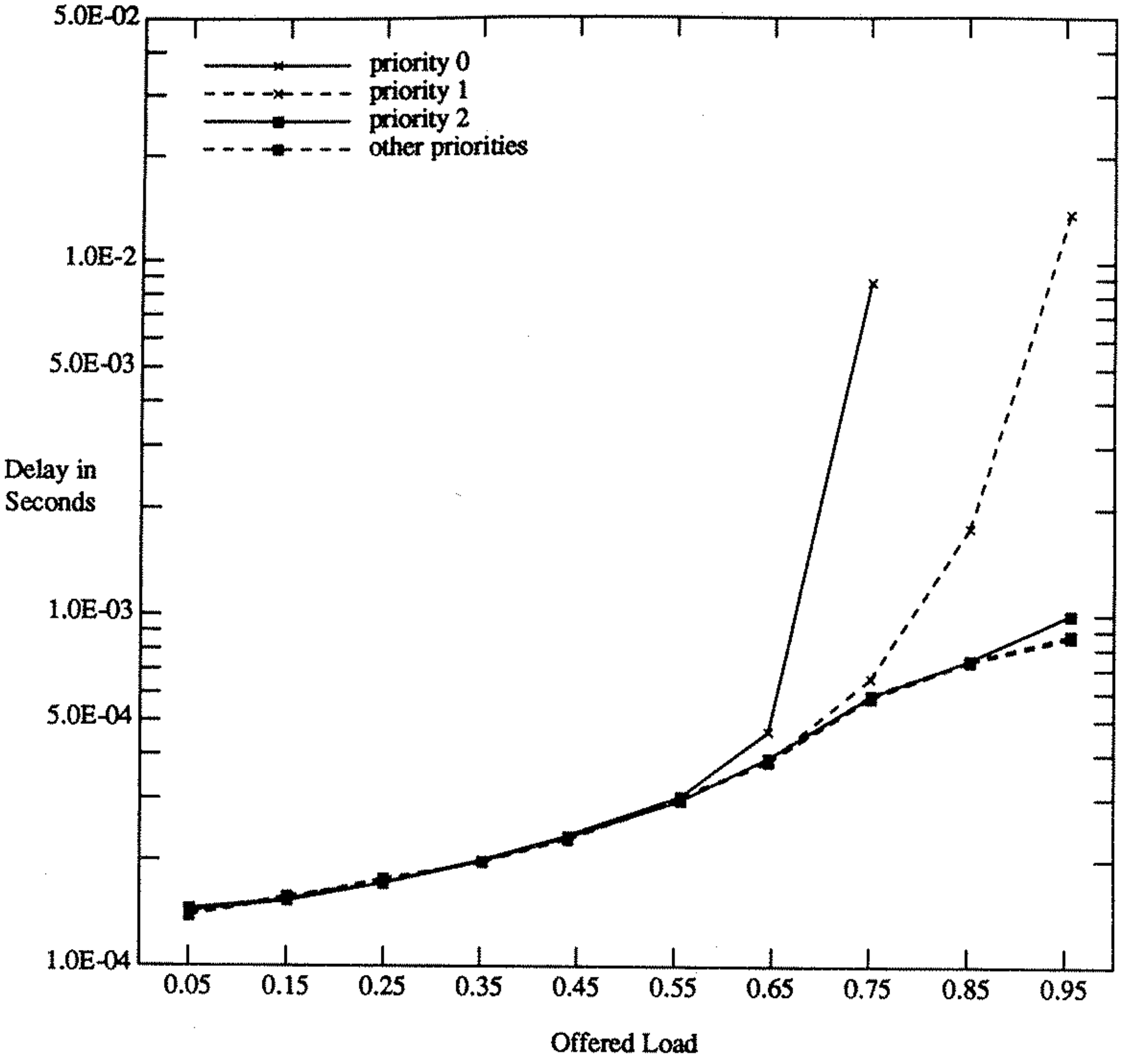

Figure 7.
Service delay, priority thresholds varied
from 1.0 ms to 4.5 ms by 0.5 ms.

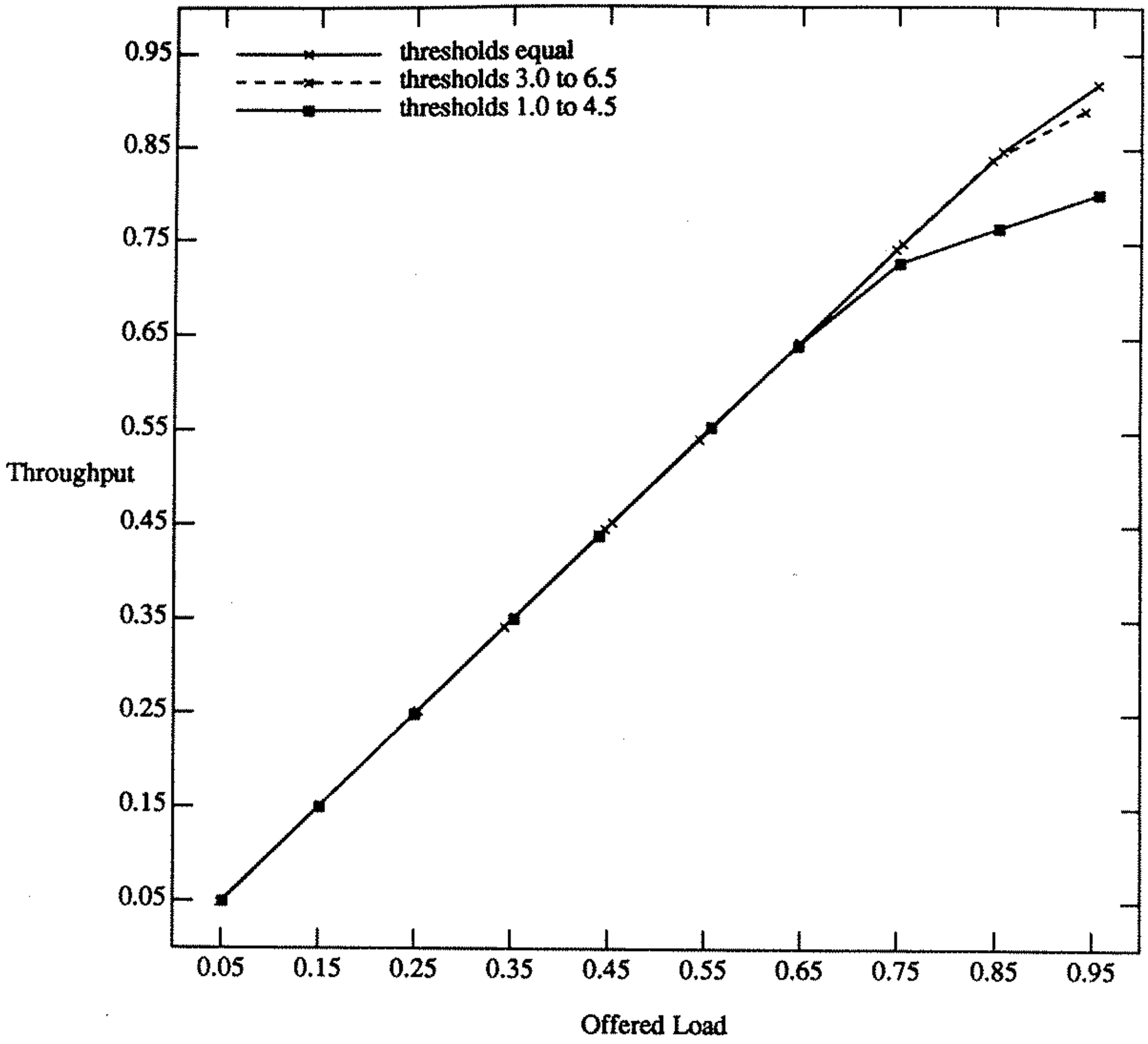

Figure 8.
Throughput, varying priority thresholds.

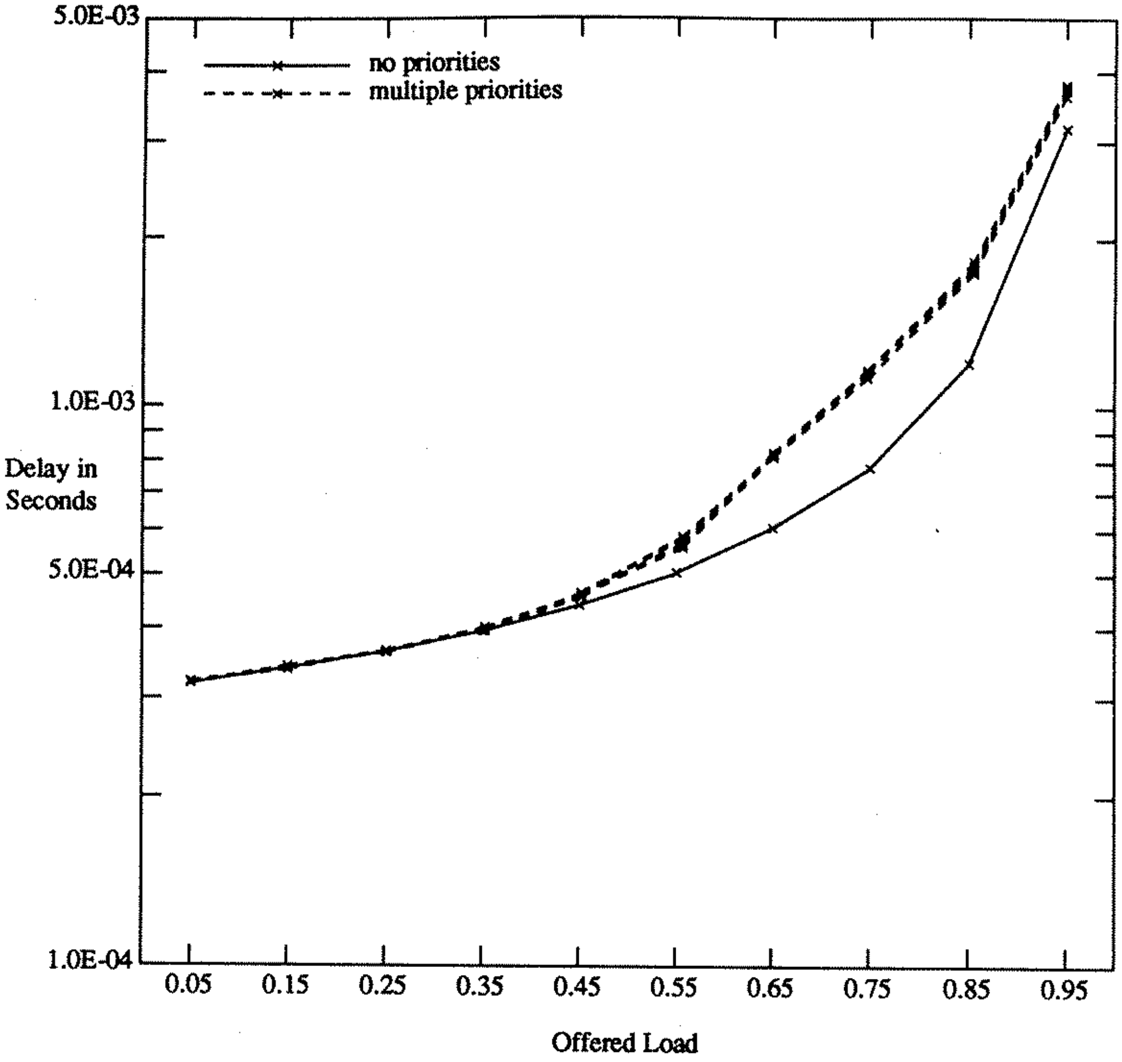

Figure 9.
Delay for single and multiple
priority operation, 40 stations.

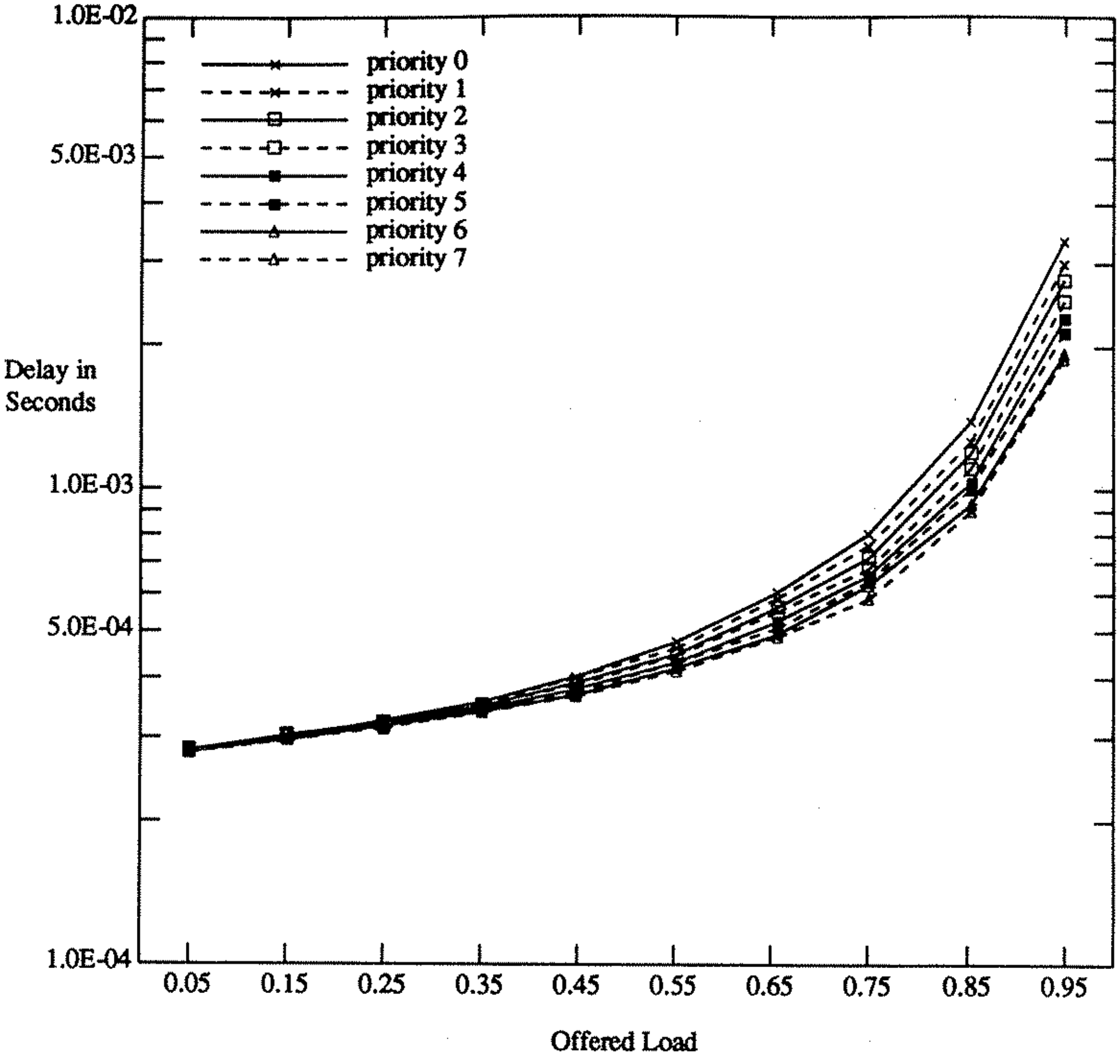

Figure 10.
Delay, 39 packets per token (default), 3 stations.