ON SCHEDULING TWO CLASSES OF REAL TIME TRAFFIC WITH IDENTICAL DEADLINES

S.PINGALI and J.KUROSE

COINS Technical Report 91-30
March 1991

On Scheduling Two Classes of Real Time Traffic With Identical Deadlines[1]

Sridhar Pingali[2] and James F. Kurose[3]

Abstract

The problem of scheduling two classes of real-time traffic with correlated time constraints is considered. Three scheduling disciplines are studied: a priority discipline which gives strict priority to one class of traffic, a threshold-based scheme in which priority is given to one class of traffic when the minimum laxity of its queued packets falls below some threshold, and a "balancing" scheme which assigns priority on the basis of the differences in minimum laxities in the two classes of traffic. Analytic results are obtained by using a discrete time model to obtain the state occupancy probabilities for the system. Here, the state is defined using the laxities of the queued real time packets. Parameters are defined to study the tradeoff in the performance of the two classes of traffic. Results are obtained to demonstrate how the balancing scheme permits us to achieve significant improvement in the performance of one class of traffic with only minimal effect on the performance of other class. A video application is suggested for this work.

1 Introduction

There is a growing realization in the networking community of the possibilities that real-time applications offer in wide area networks. Along with increasing speeds in networks, the variety of uses to which real-time services can be put is also rising. CCITT suggests a class of conversational services for broadband networks [1] that provide for bidirectional real-time communication. Examples of applications to which these services could be put would be videotelephony, video conference, voice transfer etc.

Real-time traffic operates under a deadline, i.e., a packet which is not received at its destination within a specified amount of time after its generation at a source is considered to be *lost.* The deadline itself is typically on an end-to-end basis and may be translated into local deadlines as well [7]. The performance metric of interest is thus packet loss and the goal of any good scheduling policy should be to minimize loss. Several classes of real-time traffic may arrive at a node. It then becomes the responsibility of the output multiplexer of the node to appropriately schedule packets belonging to these different streams, giving due consideration to to their relative time constraints and also their relative importance. The manner in which this is done is defined by the scheduling discipline used at the output multiplexer, and it becomes possible to compare the performance

[1]This research was supported in part by the Defense Advanced Projects Research Agency under contract NAG2-578

[2]Dept. of Electrical and Computer Engineering, UMASS, Amherst

[3]COINS Dept., UMASS, Amherst

of different scheduling disciplines with reference to appropriate metrics. In [5], for example, the Shortest Deadline First policy is shown to minimize message loss.

The problem of choosing an appropriate scheduling discipline becomes more even complex when the packets of the different classes of traffic have *correlated* time constraints. As an example, this situation could arise in video applications. There is the notion of *hierarchical source coding* which implies the separation of the digitized video signal into subsignals of differing importance [2, 3]; the relatively stable background information in a picture is separated from the information pertaining to motion. The information content of a scene, in terms of bits needed to represent it, would depend on the degree of activity in the scene. While the volume of information content could be different for these different classes of traffic, all the information pertaining to a single frame would have to be available at the same time at the receiver. Thus, the deadlines for the two classes of traffic are identical.

In this report, we consider the generic issue of scheduling different classes of traffic with the same deadline by studying three different scheduling disciplines - priority scheduling, the minimum laxity thresholding scheme and a new balancing discipline. These policies are described in Section 3. We confine our attention to two classes of traffic and provide an analytical model to study each of these scheduling disciplines. Parameters are defined that permit us to effect tradeoffs between the loss probabilities of the two classes of traffic. Our results show that the balancing scheme permits us to achieve significant improvement in the performance of one class of traffic with only a minimal sacrifice in the performance of the other class. Moreover, the balancing scheme achieves performance in which the loss of each class of traffic is often strictly less than that achievable under the thresholding policy. The analytical methodology that we use derives in part from the work presented in [4]. In [4], however, attention is given to a mixture of real-time and non-real-time traffic and hence the problem of correlated time constraints is not treated.

A study of scheduling with a view to investigate the effect of service time distributions is also given in [6]. Here too, a mixture of real-time and non-real-time traffic is considered. In [8], an analysis is carried out of the Head-of-the Line with Priority Jumps scheduling discipline for delay sensitive traffic. Again, correlations between the different classes of traffic are not considered here.

The remainder of this report is structured as follows. In Section 2, we discuss the model that we use, and in Section 3, the disciplines and metrics that we apply to this model. The solution approach that we use is considered in Section 4 and results are presented and discussed in Section 5. Section 6 concludes this report.

2 The Model

With the use of ATM networks, all classes of traffic are eventually transmitted in fixed size cells. This suggests a model of a system in which time is divided into fixed length, discrete slots, with the time needed to transmit a packet being equal to the length of the slot. We consider a multiplexer in the output buffer of a node that makes the scheduling decision with regard to two arriving classes

of real-time traffic with the same deadline. The two classes of traffic can be thought of as being queued separately and the multiplexer selects a single packet from either queue for transmission at the beginning of a slot.

We assume that the arrival streams of the two classes of traffic are independent of each other. Further, arrivals in a slot are independent of arrivals in all other slots. For each class of traffic, arrivals in a slot are treated as bulks with the bulk sizes being geometrically distributed. All arrivals in a slot are assumed to have occurred just prior to the beginning of the next slot. Each arriving packet also has an associated *laxity* equal to a prespecified deadline, τ. A packet which is not transmitted within τ slots of its arrival is considered lost. Thus, the laxity of a packet starts at τ and is reduced by one with each succeeding slot. The packet is removed from the queue if it still has not been transmitted by the time the laxity reaches zero. All arrivals in the same slot of a particular class are equivalent and all have the same chance of being picked for transmission in succeeding slots.

3 Metrics and Policies

In [4], the performance of a statistical multiplexer that schedules a mixture of real-time and non-real-time traffic was studied. The metric of interest for non-real-time traffic was average delay, and for real-time traffic it was probability of loss. Any real-time packet that does not reach its destination within the specified deadline is lost and thus the loss probability captures the performance of a particular policy for a particular class of real-time traffic. In [4], the tradeoff between the loss probability of real-time traffic and the average delay of the non-real-time traffic was investigated. Since only real-time traffic is considered in this report, the loss probabilities of the two classes of traffic may be traded off against one another by various scheduling policies; three scheduling policies are studied in this report.

3.1 Priority Discipline

In this scheduling policy, priority is always given to Class 1 traffic. Class 2 traffic is transmitted (served) only if there are no queued Class 1 cells (packets). Within a class, packets are served FCFS.

3.2 Minimum Laxity Thresholding (MLT)

The laxity of a real-time packet is the time until the expiry of that packet's deadline. As noted above, when a packet first arrives at a queue at the scheduler, its laxity is equal to its deadline and with each passing time slot, its laxity decreases by one. In the MLT discipline, a threshold is specified on the laxity of Class 1 traffic. If the minimum laxity of the queued Class 1 packets is less than or equal to the threshold, or there are no queued Class 2 packets, the minimum laxity Class 1 packet is served. The queued minimum laxity Class 2 packet is served either if the laxity

of minimum laxity Class 1 packet is greater than the threshold or if there are no queued Class 1 packets. When the threshold T is equal to the deadline, MLT becomes the same as the priority discipline. Reducing T increases the relative importance accorded to Class 2 traffic.

3.3 Balancing Discipline

In this scheduling discipline, a quantity B is specified with reference to the difference between the laxities of the minimum laxity Class 1 and Class 2 packets. A Class 1 packet is served *unless* the laxity of the minimum laxity Class 2 packet is at least B smaller than the laxity of the minimum laxity Class 1 packet. Thus, B becomes a parameter that can be varied to change the relative priorities of the two classes. When B is equal to the deadline, the balancing discipline becomes the same as the priority discipline.

As can be seen from the above descriptions, the MLT and balancing schemes provide us with parameters T and B, respectively, which can be varied to effect tradeoffs between the loss of Class 1 and Class 2 traffic. Both MLT and the balancing scheme become the same as the priority discipline in limiting cases.

4 Solution Approach

The system is modeled as a two dimensional Markov chain (x_1, x_2). Here x_1 is the laxity of the minimum laxity Class 1 packet (nominally at the head of the queue of Class 1 packets) and x_2 is the laxity of the minimum laxity Class 2 packet. This model is possible because the assumption of geometrically distributed bulk sizes with independence from slot to slot enables us to write state transition probabilities in a Markovian manner. The possible values for x_1 and x_2 are $1, 2, 3, \ldots, \tau, e$ where e represents the case of there being *no* packets of the corresponding class of traffic. Hence, there are $(\tau + 1)^2$ possible states of the system.

There exist clearly definable transition probabilities from one state to another. Due to the Markovian nature of the model, these transition probabilities depend only on the current state of the system - i.e., the probability of reaching a particular state in the following time slot depends only on the current state. Thus, we can write a matrix of state transition probabilities for the discrete time Markov chain. This matrix will be of dimensions $(\tau + 1)^2$ x $(\tau + 1)^2$. The entries in this matrix will depend on the scheduling discipline used and represent transitions from state (x_1', x_2') to state (x_1'', x_2'').

Using standard techniques, the transition matrix can be used to obtain the state occupancy probabilities in steady state. Once these state probabilities have been obtained, the throughput of each class of traffic can be obtained by simply summing probabilities over those states in which the scheduling discipline will choose that particular class of traffic for transmisson in the next slot. For example, for the priority discipline, if the system is in state $(4, 2)$, Class 1 traffic is transmitted in the next slot and thus this state contributes to throughput of Class 1 traffic. For the same discipline, if the state is $(e, 2)$, Class 2 traffic is transmitted in the next slot.

If the throughput for Class i is γ_i and the arrival rate for Class i is λ_i, then we can write the probability of loss for Class i traffic, $Ploss_i$ as

$$Ploss_i = \frac{\lambda_i - \gamma_i}{\lambda_i} \tag{1}$$

4.1 Form of the transition matrix

In order to explain how the entries for the transition matrix are obtained, we first define several quantities:

- α_i - probability of having i Class 1 arrivals in a slot
- β_i - probability of having i Class 2 arrivals in a slot
- α_{1+} - probability of at least one Class 1 arrival in a slot
- β_{1+} - probability of at least one Class 2 arrival in a slot
- p_i - probability that the laxity of the next-to-last oldest queued Class 1 packet is i slots greater than the laxity of the oldest queued Class 1 packet
- s_i - probability that the laxity of the next-to-last oldest queued Class 2 packet is i slots greater than the laxity of the oldest queued Class 2 packet
- $q_i = (1 - \sum_{j=0}^{i} p_j)$, probability that there are no Class 1 arrivals up to i slots after the arrival of the present minimum laxity Class 1 packet
- $r_i = (1 - \sum_{j=0}^{i} p_j)$, probability that there are no Class 2 arrivals up to i slots after the arrival of the present minimum laxity Class 2 packet

We noted earlier that the number of arrivals per slot of each class of traffic is geometrically distributed. Thus, it is possible to obtain each of the above quantities simply from the parameter of the geometric distribution for each class. Using all of the above, we now consider the priority case in detail. MLT and balancing schemes can be understood in an analogous fashion. We consider the specific case of the deadline τ being equal to 5. Then, the overall transition matrix will be of dimension (36 x 36). This is actually a block partitioned matrix with each block being of dimension (6 x 6). Each of these (6 x 6) matrices in turn represents the transitions of $x_1^{'}$ to $x_1^{''}$. The block location of the (6 x 6) matrix within the (36 x 36) matrix gives the values of $x_2^{'}$ and $x_2^{''}$. The basic form of the transition matrix, P^{pri} for the priority case is as follows:

$$P^{pri} = \begin{bmatrix} A_0 & A_1 & A_2 & A_3 & A_4 & R_4 \\ B_0 & B_1 & B_2 & B_3 & B_4 & R_4^{'} \\ 0 & B_0 & B_1 & B_2 & B_3 & R_3^{'} \\ 0 & 0 & B_0 & B_1 & B_2 & R_2^{'} \\ 0 & 0 & 0 & B_0 & B_1 & R_1^{'} \\ 0 & 0 & 0 & 0 & C_{1+} & C_0 \end{bmatrix}$$

Each entry in P^{pri} is a (6 x 6) matrix. The rows of P^{pri} itself correspond to values of $x_2^{'} = 1, 2, 3, 4, 5, e$ and the columns to $x_2^{''} = 1, 2, 3, 4, 5, e$. Each of the zero entries above corresponds to a (6 x 6) matrix of all zeros.

To intuitively understand why the form of P^{pri} is as shown above, let us first look at the zero entries. For example, P_{42}^{pri} is a zero matrix. This entry is for state transitions from $(x_1^{'}, 4)$ to $(x_1^{''}, 2)$ with $x_1^{'}$ and $x_1^{''}$ taking values $1, 2, \ldots, e$. Since the state indicates the laxity of the minimum laxity packet for each class, we see that it is *impossible* for the minimum laxity of Class 2 packets to drop from 4 to 2 in one slot; if a Class 1 packet is served, the minimum Class 2 laxity merely goes from 4 to 3. If the Class 2 packet is served (in the priority case, this can only happen if there are no Class 1 packets), the new minimum laxity Class 2 packet could only have arrived in the same slot as the Class 2 packet just served, or later. If it arrived in the same slot, the new Class 2 minimum laxity would be 3; if it arrived later, it would be greater than 3. Thus the probability of going from $(x_1^{'}, 4)$ to $(x_1^{''}, 2)$ is zero.

Let us now turn our attention to one of the other entries in the transition matrix, say $P_{32}^{pri} = B_0$. The form of B_0 is as follows:

$$B_0 = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 & p_4 & q_4 \\ p_0 & p_1 & p_2 & p_3 & p_4 & q_4 \\ 0 & p_0 & p_1 & p_2 & p_3 & q_3 \\ 0 & 0 & p_0 & p_1 & p_2 & q_2 \\ 0 & 0 & 0 & p_0 & p_1 & q_1 \\ 0 & 0 & 0 & 0 & s_0\alpha_{1+} & s_0\alpha_0 \end{bmatrix}$$

The first five rows of B_0 correspond to transitions $(x_1^{'}, 3)$ to $(x_1^{''}, 2)$ where $x_1^{'} = 1, 2, 3, 4, 5$. Since this is the priority discipline, the transition from 3 to 2 for x_2 happens automatically when there exists a Class 1 packet to be served. The probability of going from 2 to 1 for x_1 is p_0. This is because the current minimum laxity Class 1 packet is served and for the next minimum laxity Class 1 packet to have a laxity of 1, it should have arrived in the same slot as the packet that is currently served. The first row is the same as the second row because of the memoryless property of the geometric distribution. Similarly, we can see that all the other entries for the first five rows are correct. The last row corresponds to transitions from $(e, 3)$ to $(x_1^{''}, 2)$. Here, it is the Class 2 packet that is served and the probability of x_2 going from 3 to 2 is s_0. From an initial value of e, the value of x_1 can either go to 5 (the deadline) or remain at e. The first of these corresponds to the case of there being at least one Class 1 arrival in the current slot and the second to the case of there being no Class 1 arrivals in the current slot.

Another example is now shown. $P_{3e}^{pri} = R_3'$ is:

$$R_3' = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r_3\alpha_{1+} & r_3\alpha_0 \end{bmatrix}$$

The transition matrices for the other two schemes, MLT and balancing, are obtained by similar reasoning. The basic form of the transition matrix for the MLT case, P^{mlt} is exactly the same as that for the priority case. By examining P^{pri}, we see that there are entries that are repeated. An exactly similar repetitive structure is obtained for P^{mlt} though, the actual entries in the component matrices of P^{mlt} differ from from those of P^{pri}. For example, P_{32}^{mlt} is written for a threshold, $T = 3$, and deadline $\tau = 5$, as follows:

$$P_{32}^{mlt} = \begin{bmatrix} p_0 & p_1 & p_2 & p_3 & p_4 & q_4 \\ p_0 & p_1 & p_2 & p_3 & p_4 & q_4 \\ 0 & p_0 & p_1 & p_2 & p_3 & q_3 \\ 0 & 0 & s_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_0 & 0 & 0 \\ 0 & 0 & 0 & 0 & s_0\alpha_{1+} & s_0\alpha_0 \end{bmatrix}$$

It is instructive to compare this with the corresponding entry, B_0, in P^{pri}. It can be seen that the first three rows of the two matrices are the same. This is because up to the value of the threshold, priority scheduling and MLT behave in the same fashion. The last row is also the same because that corresponds to $x_1' = e$, i.e., there are no Class 1 packets that may be served in this slot. Rows 4 and 5 are different in accordance with the manner in which MLT differs from the priority discipline.

The form of the transition matrix for the balancing discipline is a little more difficult to obtain. Since it is *difference* between x_2' and x_1' that counts in determining which Class of traffic is served, care must be taken while filling the transition matrix. It must always be kept in mind that the case of either or both of the queues corresponding to each class of traffic (i.e., either $x_1' = e$ and/or $x_2' = e$) is a special case and must be treated accordingly. For the specific case of $\tau = 5$ and $B = 3$, we may write a general form for the transition matrix in the following way:

$$P^{bal} = \begin{bmatrix} D_0 & D_1 & D_2 & D_3 & D_4 & S_4 \\ E_0 & E_1 & E_2 & E_3 & E_4 & S_4' \\ 0 & F_0 & F_1 & F_2 & F_3 & S_3' \\ 0 & 0 & F_0 & F_1 & F_2 & S_2' \\ 0 & 0 & 0 & F_0 & F_1 & S_1' \\ 0 & 0 & 0 & 0 & G_{1+} & G_0 \end{bmatrix}$$

For this specific case, the first row is exactly the same as for the MLT case for a value of $T = 3$. This is so because Class 1 traffic is served unless Class 2 minimum laxity is atleast 3 less than Class 1 minimum laxity. For the first row, Class 2 minimum laxity (x_2') is 1. Hence, Class 1 traffic is served for $x_1' = 1, 2, 3$ and Class 2 traffic for $x_1' = 4, 5, e$. The last four rows (corresponding to $x_2' = 3, 4, 5, e$) are exactly the same as for the priority case. This happens because for higher values of x_2', x_1' never gets to be atleast B larger than x_2'. The second row behaves in a manner particular to the balancing discipline. The value of x_2' is 2 and hence Class 1 traffic is served for $x_1' = 1, 2, 3, 4$. A sample entry from this row is shown below:

$$P_{23}^{bal} = E_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & s_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & s_2\alpha_{1+} & s_2\alpha_0 \end{bmatrix}$$

5 Results

The method described in the previous section was used to develop the transition probability matrices for each of the three cases. Standard numerical techniques were used to solve the matrix equations for state occupancy probabilities in steady state. This was done in each case for the specific deadline value of 10. Once these probabilities are obtained, the throughput for each class of traffic is obtained by simply summing the appropriate state probabilities. For instance, in the priority case, throughputs for the two classes of traffic can be written as

$$\gamma_1^{pri} = \sum_{i=1}^{\tau} \sum_{j=e,1}^{\tau} P(i,j) \tag{2}$$

and

$$\gamma_2^{pri} = \sum_{j=1}^{\tau} P(e,j) \tag{3}$$

where $P(i,j)$ gives the state occupancy probability for the system state (i,j) in steady state. Once the throughputs are obtained, the respective probabilities of loss can be obtained using Equation 1.

Throughput equations for the MLT case are given by

$$\gamma_1^{mlt} = \sum_{i=1}^{T} \sum_{j=e,1}^{\tau} P(i,j) \tag{4}$$

and

$$\gamma_2^{mlt} = \sum_{i=e,T+1}^{\tau} \sum_{j=1}^{\tau} P(i,j) \tag{5}$$

Throughput equations for the balancing case are

$$\gamma_1^{bal} = \sum_{(i-j)<B, j=e, i\neq e} P(i,j) \tag{6}$$

$$\gamma_2^{bal} = \sum_{(i-j)\geq B, i=e, j\neq e} P(i,j) \tag{7}$$

Equation 6 arises from the fact that Class 1 traffic is served in the balancing case whenever the minimum Class 1 laxity is either less than the minimum Class 2 laxity, or if minimum Class 1 laxity is utmost $(B-1)$ greater than minimum Class 2 laxity. Both these situations are taken care of by the condition $(i-j) < B$ in the summation in Equation 6. The other two conditions on the summation arise because Class 1 traffic is served whenever there are no Class 2 packets (i.e., $j = e$), *provided* there are some Class 1 packets waiting for service (i.e., $i \neq e$).

Figures 1 through 4 show the plots for $Ploss_1$ vs $Ploss_2$ for $\tau = 10$ for the case of balanced traffic. These figures look at arrival rates (expressed as the average bulk size/slot) of 0.3, 0.4, 0.45 and 0.5. The arrival rates are the same for each class of traffic (i.e., $\lambda_1 = \lambda_2$). $Ploss_1$ (the Y-axis) is plotted on a log scale. For the MLT and balancing disciplines, the parameters T and B can respectively be varied to yield a set of achievable performance levels. For the priority discipline, there is no such parameter and there is only one point in the graph for each value of the arrival rate. In all the figures, T and B go from a value of 1 to a value of τ (i.e., the deadline). Both T and B increase with increasing X-axis values. As can be seen, both MLT and balancing tend to the priority case in the limiting cases of $T = \tau$ and $B = \tau$, respectively, and all three disciplines have identical performance. The lines joining the points are indicated for clarity - the points themselves are obtained through independent solutions to the set of state transition equations.

As can be seen from the graphs, as T increases, $Ploss_1$ decreases in the MLT case. This happens because the chance of serving Class 1 increases. For the balancing case, as B increases, $Ploss_1$ decreases. This is true because as B increases, Class 2 minimum laxity has to be much smaller than Class 1 minimum laxity for a Class 2 packet to be served. Consequently, the chance of serving Class 1 increases thereby reducing loss probability for Class 1.

Not unexpectedly, the probability of loss for both classes increases when the arrival rates increase. This can be seen by simply comparing the numbers across graphs for increasing arrival rate. A more interesting result can be obtained by comparing the MLT and balancing curves on the same graph. It is clear that the balancing scheme achieves a lower maximum loss (for *both* classes) over a range of T and B values. Since T and B are independent parameters, they cannot strictly be compared. However, no matter what value is chosen for T, it is possible to choose a B such that the probability of loss of Class 1 traffic is lower in the balancing case than in the MLT case and that of Class 2 traffic is not substantially higher for the balancing case than for the MLT case. However, it must be pointed out that the MLT case can always attain a lower minimum value of $Ploss_2$ than the one achievable by the balancing case. This is made possible only at the expense of high values for $Ploss_1$.

As the rate increases, the two curves come closes together. However, for rates of 0.3 and 0.4 (corresponding to offered loads of 0.6 and 0.8, respectively), the balancing graph is more flat than the MLT scheme. Thus, the parameter B can be used to effect a useful tradeoff between $Ploss_1$ and $Ploss_2$. By reducing B, we move towards the left along the balancing curve and increase the relative importance given to Class 2 traffic. By looking at Figure 1, we can see that a halving of $Ploss_2$ is possible while still keeping $Ploss_1$ at a low level. At $B = T = \tau = 10$, MLT and balancing give the same performance with $Ploss_1 = 0.000004$ and $Ploss_2 = 0.011537$. At $T = 8$, the MLT case gives $Ploss_2 = 0.006037$, but with $Ploss_1 = 0.000018$. Thus, a 48% reduction in $Ploss_2$ is achieved at the cost of a 350% increase in $Ploss_1$. For the balancing case, however, we obtain $Ploss_2 = 0.006313$ and $Ploss_1 = 0.000005$ for $B = 7$. Therefore, a 25% increase in $Ploss_1$ is sufficient to achieve a 45% reduction in $Ploss_2$. In fact, at $B = 8$, we get $Ploss_2 = 0.008157$ with no change in $Ploss_1$ (i.e., $Ploss_1 = 0.000004$). Thus a 29% drop in $Ploss_2$ is possible with 0% increase in $Ploss_1$. The MLT scheme, on the other hand, demands a 100% increase in $Ploss_1$ (to 0.000008) for $T = 9$, while providing a 28% drop in $Ploss_2$ (to 0.008360). The performance of the balancing scheme is clearly better than that of MLT for operation in this region.

Figures 5 to 8 look at the case of unbalanced traffic for a deadline of $\tau = 10$. Figures 5 and 7 are plotted on linear scales on both axes because the loss probability of Class 1 traffic drops to zero as the priority accorded Class 1 traffic increases while the rate at which Class 1 traffic arrives is maintained at only a fifth that of Class 2 traffic. Figures 6 and 8 are plotted with $Ploss_1$ on a log scale as before. Here, the arrival rate of Class 2 traffic is a fifth that of Class 1 traffic. The conclusions drawn above with regard to balanced traffic are still largely valid. For lower overall load (Figures 5 and 6), the balancing scheme still performs well. By suitably choosing the value of B, a substantial decrease in $Ploss_2$ can be obtained without an increase in $Ploss_1$ as shown in Figure 6. This is achieved despite the fact that Class 1 traffic arrives at five times the rate of Class 2 traffic. However, when the overall load is high, as in Figure 8, a similar claim cannot be made. The reason for this is clear intuitively.

Figure 9 looks at the case of a much more relaxed deadline constraint of $\tau = 20$ for a balanced load. What is interesting is that $Ploss_1$ quickly approaches 0 for all the scheduling disciplines for a not too high a value of $Ploss_2$. This suggests, quite reasonably, that scheduling becomes an important problem in the context of tight time constraints. At more relaxed constraints, it might be more meaningful to look at other metrics - for example, average delay - and study the effect of various scheduling disciplines on those performance measures.

6 Conclusions

In this report, we have considered the problem of scheduling two classes of real-time traffic with correlated time constraints. Three scheduling disciplines were studied: a priority discipline which gives strict priority to one class of traffic, a threshold-based scheme in which priority is given to one class of traffic when the minimum laxity of its queued packets falls below some threshold, and

a "balancing" scheme which assigns priority on the basis of the differences in minimum laxities in the two classes of traffic. Our analytic results showed that the balancing discipline, which explicitly considers the difference between minimum laxities of the two classes of traffic, can yield both better performance for *both* classes of traffic than MLT and can be more effectively used to exploit the tradeoffs that exist between the two classes of traffic. This was found to be particularly the case when time constraints were relatively tight and links were loaded up to an 80 percent nominal load.

In carrying out the analysis, we made several assumptions. Since we were interested in the fundamental problem of scheduling two classes of real-time traffic with correlated time constraints, these helped make the analysis easier. For particular applications, however, these assumptions might be hard to justify. There is the assumption of independence between the arrivals of the two classes. With reference to video applications, since both classes are obtained from the same video source, the chances are that there is correlation between them. Further, we assume independence from slot to slot. Again, for video applications, this is difficult to justify. Usually, what is transmitted over the network is the *difference* in the picture from frame to frame with a regular refresh of the entire picture [2]. Hence, there is likely to be correlation from slot to slot. Thus a logical extension of the work would be to relax these independence assumptions and to carry out the analysis again. It is to be expected, however, that this would be a more difficult task and that we would have to look at a different state description for the system.

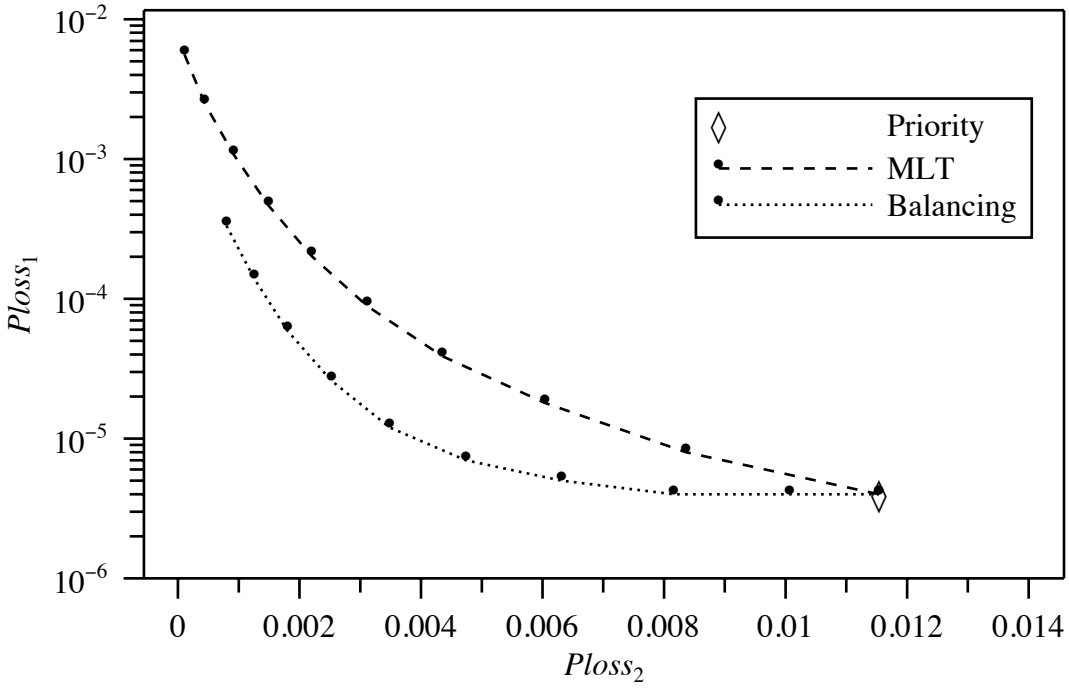

Figure 1: $\lambda_1 = \lambda_2 = 0.3, \tau = 10$

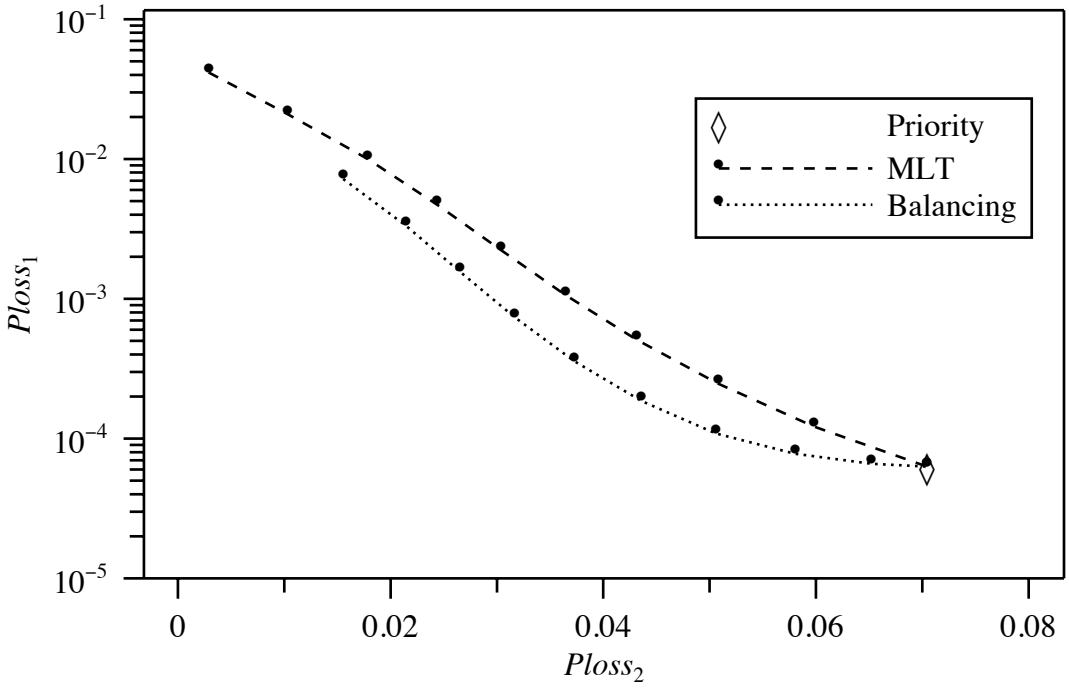

Figure 2: $\lambda_1 = \lambda_2 = 0.4, \tau = 10$

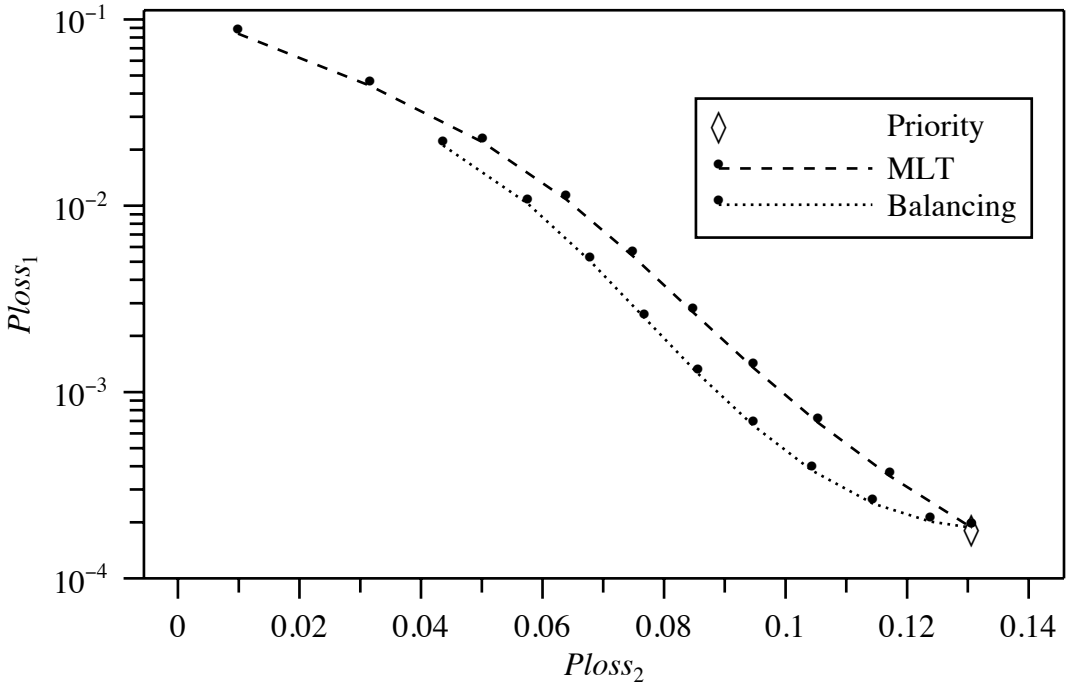

Figure 3: $\lambda_1 = \lambda_2 = 0.45, \tau = 10$

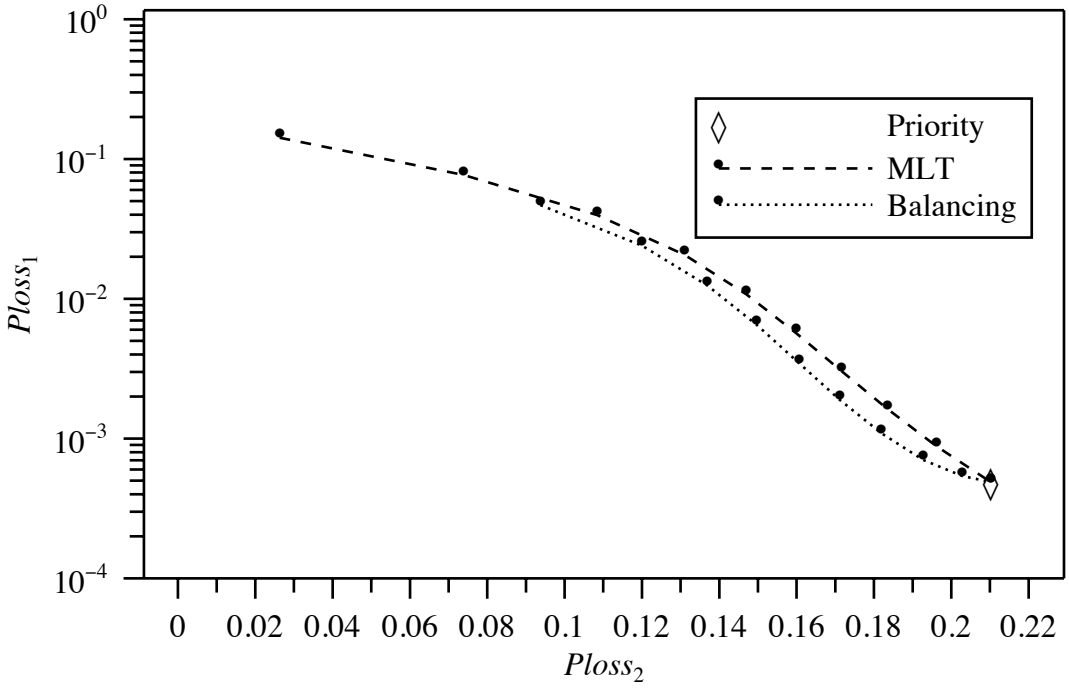

Figure 4: $\lambda_1 = \lambda_2 = 0.5, \tau = 10$

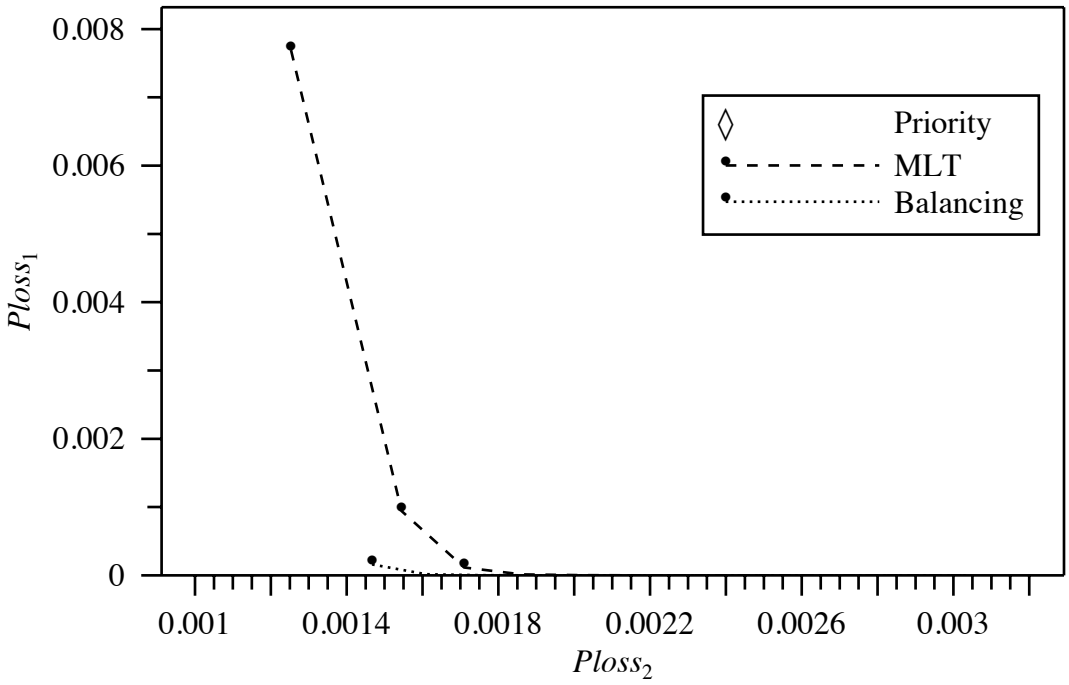

Figure 5: $\lambda_1 = 0.1, \lambda_2 = 0.5, \tau = 10$

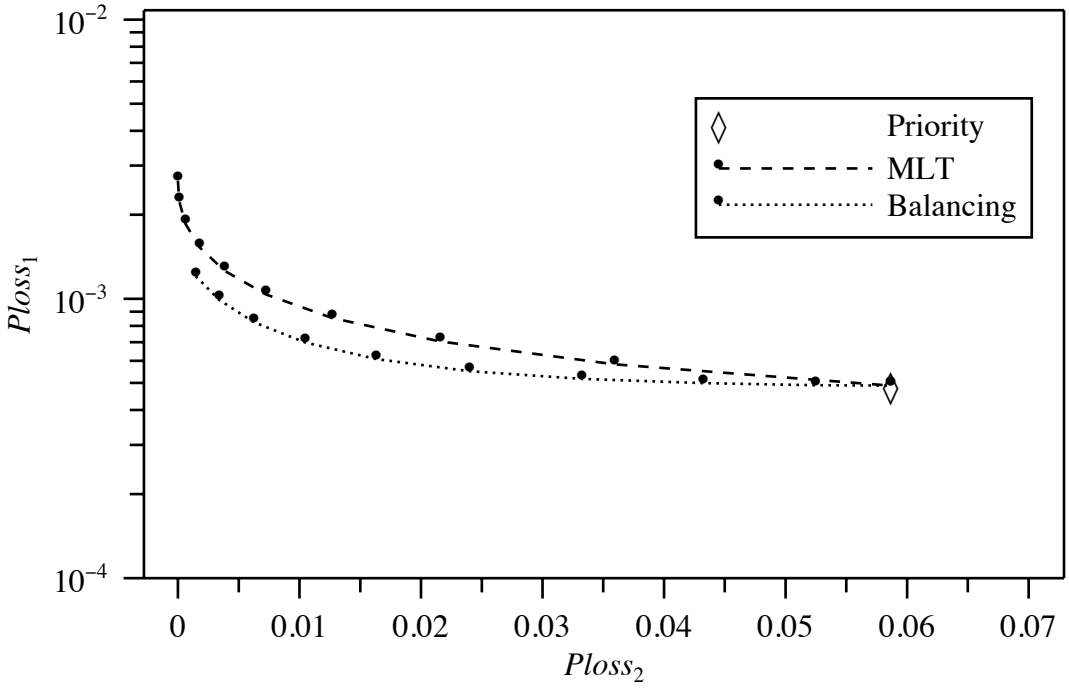

Figure 6: $\lambda_1 = 0.5, \lambda_2 = 0.1, \tau = 10$

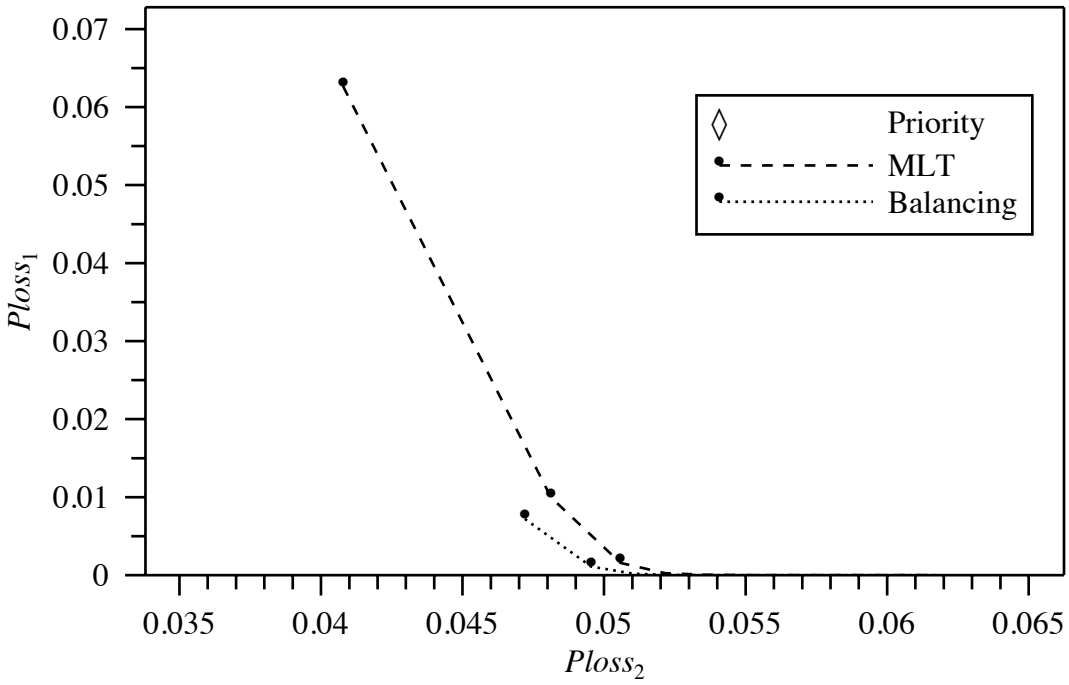

Figure 7: $\lambda_1 = 0.15, \lambda_2 = 0.75, \tau = 10$

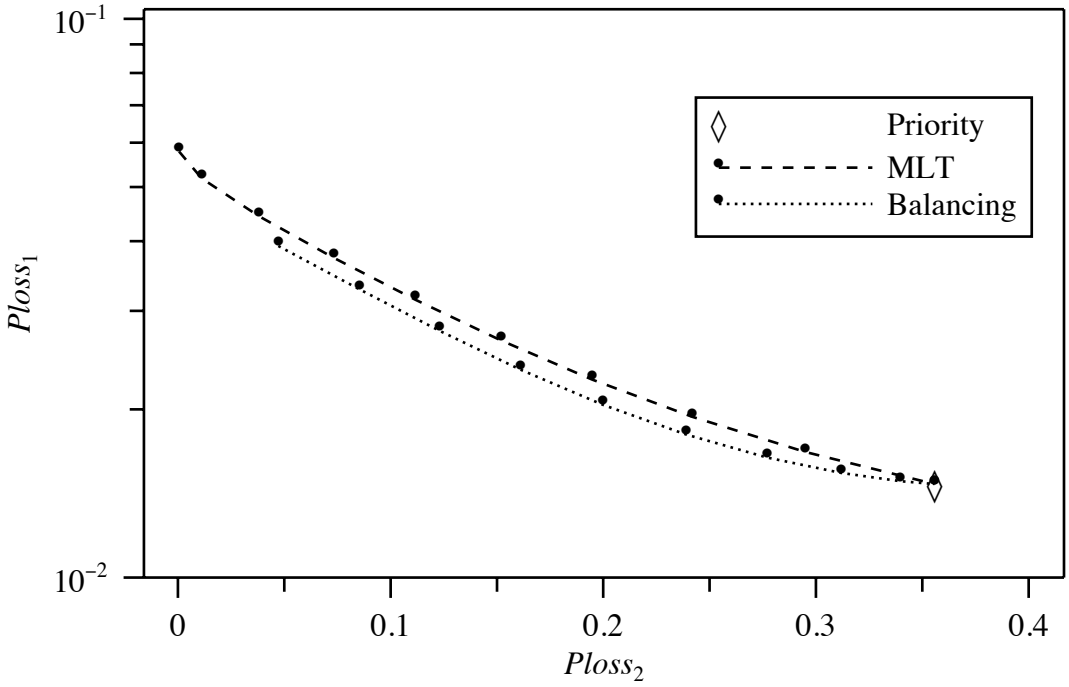

Figure 8: $\lambda_1 = 0.75, \lambda_2 = 0.15, \tau = 10$

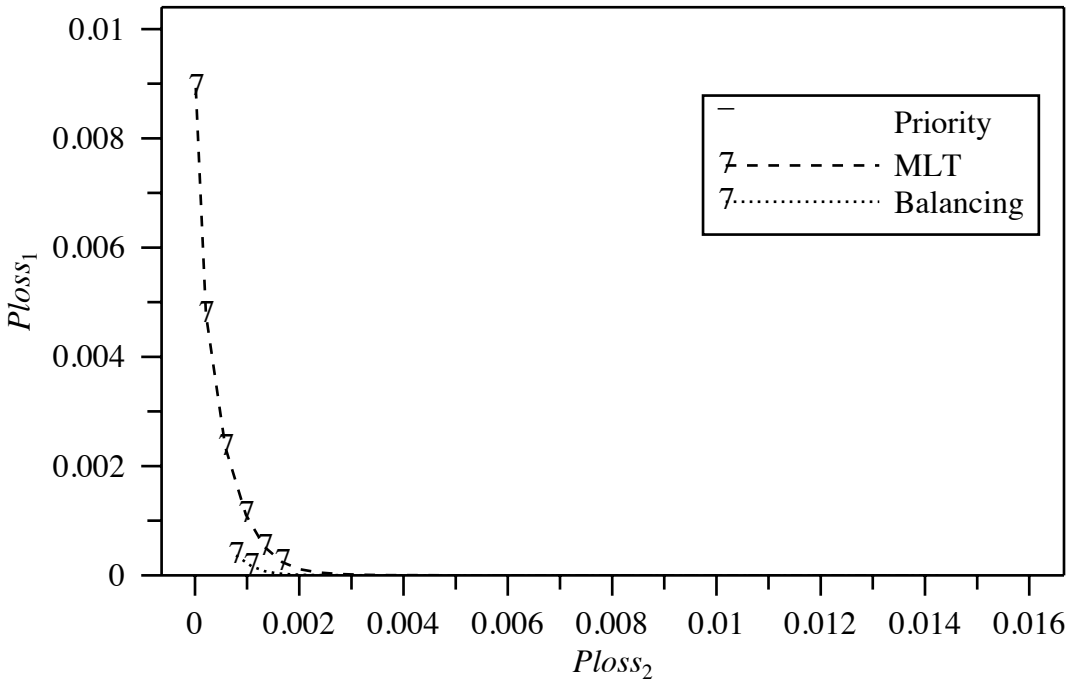

Figure 9: $\lambda_1 = \lambda_2 = 0.4, \tau = 20$

References

[1] CCITT Study Group XVIII, Report R 34, I.211 B-ISDN service aspects (Draft Recommendation), June 1990.

[2] G.Karlsson, M.Vetterli, "Packet Video and Its Integration into the Network Architecture", *IEEE, J. Select. Areas Commun.*, pp 739-751, June 1989.

[3] M.Ghanbari, "Two-Layer Coding of Video Signals for VBR Netwroks", *IEEE, J. Select. Areas Commun.*, pp 771-789, June 1989.

[4] R.Chipalkatti, J.F.Kurose, D.Towsley, "Scheduling Policies for Real-Time and Non-Real-Time Traffic in a Statistical Multiplexer", *Proc. IEEE Infocom '89*, Ottawa, April 1989.

[5] S.S.Panwar, "Time Constrained and Multi-access Communication", Ph.d Thesis, 1986, Dept. of Electrical and Computer Eng., Univ. of Mass. Amherst, MA 01003.

[6] V.Ramaswami, D.D.Lucantoni, "Algorithmic Analysis of a Dynamic Priority Queue", in R.L.Disney and T.J.Ott (eds.), *Applied Probability - Computer Science: The Interface, Vol. II*, (Birkhauser, Boston MA), pp. 157-206, 1982.

[7] H. Schulzrinne, J.F. Kurose and D. Towsley, "Congestion Control by Selective Packet Dropping for Real-Time Traffic in High-Speed Networks," *Proc IEEE Infocom '90,* pp. 543-550, San Francisco, June 1990.

[8] Y.Lim, J.Kobza, "Analysis of a Delay-Dependent Priority Discipline in a Multi-Class Traffic Packet Switching Node", *Proc. IEEE Infocom '88*, New Orleans, March 1988.