

Deadline-Based Scheduling in Support of Real-Time Data Delivery

Yanni Ellen Liu*

Department of Computer Science
University of Manitoba

Winnipeg, Manitoba, Canada R3T 2N2

ylliu@cs.umanitoba.ca

Johnny W. Wong

School of Computer Science
University of Waterloo

Waterloo, Ontario, Canada N2L 3G1

jwwong@uwaterloo.ca

Abstract

The use of deadline-based scheduling in support of real-time delivery of application data units (ADUs) in a packet-switched network is investigated. Of interest is priority scheduling where a packet with a smaller ratio of T/H (time until delivery deadline over number of hops remaining) is given a higher priority. We refer to this scheduling algorithm as the T/H algorithm. T/H has time complexity of $O(\log N)$ for a backlog of N packets and was shown to achieve good performance in terms of the percentage of ADUs that are delivered on time. We develop a new and efficient algorithm, called T/H- p , that has $O(1)$ time complexity. The performance difference of T/H, T/H- p and FCFS are evaluated by simulation. Implementations of T/H and T/H- p in high-speed routers are also discussed. We show through simulation that T/H- p is superior to FCFS but not as good as T/H. In view of the constant time complexity, T/H- p is a good candidate for high-speed routers when both performance and implementation cost are taken into consideration.

Keywords:

Real-time data delivery, deadline-based scheduling, packet-switched networks, performance evaluation

*Corresponding author, telephone: 1-204-474-9740, fax: 1-204-474-7609

1 Introduction

Many applications running on the Internet require timely delivery of real-time data. Examples of real-time data include stock quote updates and stock trades in on-line trading, bids in an on-line auction, state updates in a distributed multi-player on-line game, time-sensitive business documents in electronic commerce applications, video and audio data in a video conference, and voice data in IP telephony. To ensure timely delivery of real-time data, quality of service (QoS) support at the transport network is required.

Techniques for QoS support include fair queueing and buffer management at routers [1, 2], integrated services and resource reservation [3, 4], and differentiated services [5]. These strategies are based on the notion of a *flow*; each flow specifies its bandwidth and performance requirements, and QoS support is provided on a per flow basis. For certain types of real-time data transfer, such as the delivery of a small document (*e.g.*, a stock quote update), it might be difficult to specify a flow with given bandwidth requirements because data will not be presented to the network for a significant period of time.

In [6], a deadline-based network resource management framework in support of real-time document delivery was developed. In that framework, a document is viewed as an application data unit (ADU); it may correspond to a file or to a frame in audio or video transport. Each ADU is associated with a delivery deadline, which is a time-of-day measure that represents the absolute time at which this ADU should be delivered at the receiver. This application-layer deadline is mapped onto a network-layer deadline, which is carried by packets and used by routers for channel scheduling purpose. Deadline-based channel scheduling is employed at the network layer; the objective is to maximize the percentage of ADUs that are delivered on time.

Deadline-based scheduling algorithms have been studied in the context of job scheduling [7, 8]. In [7], a network of servers is considered and an algorithm that selects the job with the smallest ratio T/H at each server, where T is time until deadline and H is number of hops remaining, was shown to perform well in terms of the percentage of jobs that are completed on-time. We will refer to this algorithm as T/H . T/H has also been studied in the context of packet-switched networks [9, 10, 11].

The T/H algorithm has time complexity $O(\log N)$ for a backlog of N packets. Its efficient implementation in high-speed routers requires hardware designs [12, 13, 14] where the resources required increases with the number of priority levels, or in the context of T/H, the number of dissimilar T/H values associated with packets to be processed.

In [6], the authors proposed a variant of the T/H algorithm called T/H-L. It is based on the use of a threshold L . Packets with $T/H \leq L$ have higher priority than those with $T/H > L$. T/H-L is implemented by two head-of-the-line priority queues. For a given L , the time complexity of T/H-L is $O(1)$. In [6], simulation results were presented to compare the performance of T/H, T/H-L, and FCFS which is typically used in routers that provide a best-effort service. These results showed that (i) T/H has the best performance in terms of the percentage of ADUs that are delivered on-time, (ii) T/H-L is superior to FCFS if the threshold L is chosen properly, and (iii) there is a wide range of values of L for which the performance of T/H-L is close to that of FCFS. The effectiveness of T/H-L therefore relies on the ability to determine a good value of L efficiently. A procedure to determine the best value of L was presented in [15]. However, that procedure is based on an empirically derived relationship; it is rather complicated and is not practical in terms of implementation in high-speed routers.

In this paper, a new variant of T/H, called T/H- p , is developed. This algorithm also has time complexity $O(1)$. In its simplest form, two head-of-the-line priority queues are defined; packets in the higher priority queue have smaller T/H values. Incoming packets are forwarded to the two queues such that the fraction of packets joining the higher priority queue is p . Our investigation shows that (i) it is possible to determine a value of p such that T/H- p is superior to FCFS over a range of operating conditions and (ii) efficient implementation in high speed routers is possible and less costly than that for T/H. The remark about efficient implementation is supported by the following two observations. First, the low time complexity of T/H- p allows for hardware implementations that are not affected by the number of dissimilar T/H values of packets to be processed. Secondly, an efficient procedure can be developed to realize the fraction p of packets that should enter the high priority queue. This procedure is based on adjusting a threshold L (similar to that in T/H-L) dynamically using parameters that can be obtained easily. Our work on T/H- p therefore represents a

significant advancement to the work on T/H-L reported in [15].

Our investigation also shows that T/H has better performance than T/H- p . However, in view of the difference in time complexity, T/H- p is an attractive algorithm for high-speed routers when both performance and complexity are taken into consideration.

This paper is organized as follows. Section 2 reviews literature on deadline-based scheduling. Section 3 presents the design and implementation of our new deadline-based scheduling algorithm T/H- p . The performance of T/H, T/H- p , and FCFS is evaluated by simulation in Section 4. Practicality of the T/H and T/H- p algorithms is discussed in Section 5. Extension of the T/H- p algorithm to more than two queues is described in Section 6. Finally, Section 7 contains a summary of our findings.

2 Related Work

Deadline-based scheduling has been used to schedule tasks or processes in real-time operating systems on a single processor. In [16], a preemptive earliest due-date first (EDF) algorithm was proved to be optimal in the sense that if a set of periodic hard real-time tasks can be scheduled by any algorithm, it can be scheduled by EDF. EDF and its variants have also been used to provide network-layer QoS guarantees [17, 18]. Within the context of providing deterministic QoS guarantees, a multiple FIFO queue based constant time algorithm approximating the EDF scheduler was presented in [19].

The idea of specifying a lifetime for voice packets and performing channel scheduling based on packet lifetime has been investigated. In [9], a simulation study was conducted to compare the performance of FCFS and the following two channel scheduling algorithms:

- (1) Delay-dependent voice (DDV): this is a head-of-the-line discipline [20] with voice data having the highest priority. Voice packets are serviced using a deadline-based algorithm. At any time instant, packets with smaller values of $(LT - A)/H$ have higher priority, where LT is an end-to-end packet lifetime, A is the elapsed time until start service since the packet has entered the network, and H represents the number of hops remaining.

- (2) Head-of-line voice (HOLV): this is also a head-of-the-line discipline. There are three queues. When a voice packet arrives at a channel, a recommended maximum delay given by $Rm = (LT - A(r))/H$ is calculated, where $A(r)$ is the elapsed time until arrival at the channel since the packet has entered the network. Voice packets enter the highest or medium priority queue depending on whether their Rm values are smaller than a threshold or not. Small data packets enter the medium priority queue, while large data packets enter the queue with the lowest priority.

It was found that DDV yields the best performance in terms of the percentage of voice packets that are delivered on-time, and HOLV has comparable performance [9].

In [10, 11], the performance of the following four channel scheduling algorithms were investigated:

- (1) Transit priority (TP): transit traffic has higher priority than originating traffic.
- (2) FIFO+ [21]: for a packet arriving at a node, the delay at this node (denoted by D) and the average delay experienced by packets in the same class at this node (denoted by D_c) are measured. The excess delay, given by $D - D_c$ is accumulated and carried by the packet. The packet with the largest excess delay is serviced first.
- (3) Hop laxity (HL): at the i -th hop, the packet with the smallest value of $(d + M - A)/(M - i + 1)$ is serviced next, where d is the packet deadline, M is the total hop count between source and destination, and A is the age of the packet.
- (4) Minimum laxity (ML): the priority of a packet is determined by $d + M - A$.

Simulation results showed that HL had the best performance in terms of the fraction of packets that are delivered on-time. These results were confirmed with an implementation of HL in a BSD-based kernel and its experimentation on the DARTnet network [11].

Due-date based scheduling was also studied in the context of job-shop scheduling. A simulation study of a randomly routed job-shop process was reported in [7]. Besides FCFS, EDF, and shortest processing first (SPT), three other algorithms were investigated:

- (1) Smallest operation due-date first: the operation due-date is given by (due-date – job arrival time at the shop) / number of operations.
- (2) Smallest slack first: slack is given by (time until deadline – remaining processing time).
- (3) Slack over number of operations (S/OPN): the job with the smallest ratio of slack over number of operations remaining has the highest priority.

It was found that S/OPN and SPT yield the best performance in terms of the percentage of jobs that are completed on-time. Between the two, S/OPN has better performance when the offered load is less than 90%; SPT becomes superior when the load is beyond 90%.

3 The T/H- p Algorithm

The T/H- p algorithm is depicted in Figure 1. There are three priority queues. Two classes

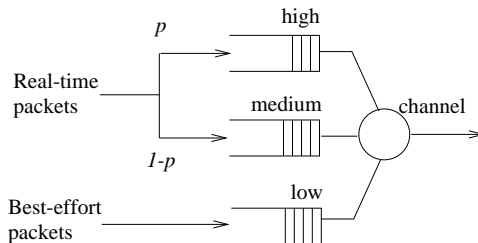


Figure 1: Queueing model for T/H- p

of packets are considered: real-time and best effort. Best effort packets do not have delivery deadline; they always join the low priority queue. For the real-time packets, a fraction p ($0 \leq p \leq 1$) of the more urgent real-time packets (or packets with smaller T/H values) join the high priority queue, and the remaining fraction of $(1 - p)$ joins the medium priority queue. When the channel becomes idle, it serves the high priority queue if that queue is nonempty, otherwise it serves the medium priority queue. The low priority queue is served only when both high and medium priority queues are empty. The scheduling discipline at each of the three queues is FCFS. To implement T/H- p , there are two questions that need to be answered. First, what value of p should be used? Second, for a given value of p , how does

a scheduler decide whether a real-time packet should join the high priority or the medium priority queue? These questions will be addressed in this section.

3.1 Selection of p

We note that for a packet with a delivery deadline, the time left can be judiciously amortized to a per-hop-slack at individual hops. If this packet is transmitted within the slack at each hop, then the packet will arrive at the receiver before its deadline. It follows that maximizing the on-time performance at each hop would have a positive impact on the overall on-time performance. Based on this observation, we use a single channel analytic model to determine p that would result in good performance for a range of traffic conditions.

3.1.1 Model of a Single Channel

Our model is an M/G/1 queueing model with head-of-the-line priority [22]. Focusing on real-time packet only, this model is shown in Figure 2. Real-time packets are assumed to

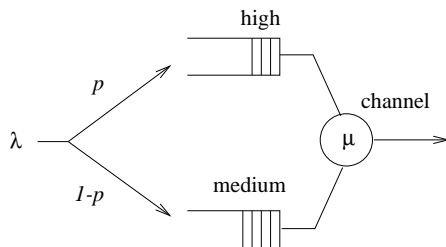


Figure 2: Queueing model for T/H- p (real-time queues)

arrive at the server according to a Poisson process with rate λ . A fraction p of the packets join the high priority queue (or queue 1); these packets belong to class 1. The remaining fraction $(1 - p)$ joins the medium priority queue (or queue 2) and belongs to class 2. Packets are served using the head-of-the-line priority discipline, and the service time may follow an arbitrary distribution.

3.1.2 Analytic Results

Let λ_1 and λ_2 be the arrival rates of class 1 and class 2 packets respectively. Assuming that the T/H values of real-time packets are independent of the arrival process, we have $\lambda_1 = \lambda p$ and $\lambda_2 = \lambda (1 - p)$. Let $f_{\tilde{d}}(t)$ be the probability density function (pdf) of the T/H value of a packet. One can then determine a value j such that $Prob[\text{T/H value of a packet} \leq j] = p$ by solving:

$$\int_0^j f_{\tilde{d}}(t) dt = p. \quad (1)$$

One can also obtain the pdfs of \tilde{d}_1 and \tilde{d}_2 , the T/H values for class 1 and class 2 packets. They are:

$$f_{\tilde{d}_1}(t) = \frac{f_{\tilde{d}}(t)}{p}, \quad 0 < t \leq j \quad (2)$$

and

$$f_{\tilde{d}_2}(t) = \frac{f_{\tilde{d}}(t)}{1-p}, \quad j < t < \infty. \quad (3)$$

Let \tilde{s}_1 and \tilde{s}_2 be the sojourn times of class 1 and class 2 packets respectively. The percentage of real-time packets that are on-time (denoted by q_s) can be computed from

$$q_s = p \times Prob[\tilde{s}_1 \leq \tilde{d}_1] + (1-p) \times Prob[\tilde{s}_2 \leq \tilde{d}_2] \quad (4)$$

where $Prob[\tilde{s}_1 \leq \tilde{d}_1] = \int_0^j Prob[\tilde{s}_1 < t] f_{\tilde{d}_1}(t) dt$ and $Prob[\tilde{s}_2 \leq \tilde{d}_2] = \int_j^\infty Prob[\tilde{s}_2 < t] f_{\tilde{d}_2}(t) dt$.

For the model in Figure 2, analytic results for the sojourn time distributions are available in [22]. Let \tilde{x}_i be the service time of class i packets, \bar{x}_i be its mean, and $B_i^*(s)$ be the Laplace transform of its pdf, $i = 1, 2$. The traffic intensity of class i , denoted by ρ_i , is equal to $\lambda_i \bar{x}_i$. Let $S_i^*(s)$ be the Laplace transform of the pdf of the sojourn time \tilde{s}_i . For class 1, $S_1^*(s)$ is given by [22]:

$$S_1^*(s) = \left(\frac{1 - \rho_1 - \rho_2}{1 - \rho_1} + \frac{\rho_2 B_{2r}^*(s)}{1 - \rho_1} \right) \frac{(1 - \rho_1) B_1^*(s)}{1 - \rho_1 B_{1r}^*(s)}, \quad (5)$$

where $B_{ir}^*(s)$ is the Laplace transform of the distribution of the residual life of class i . $B_{ir}^*(s)$ is given by [23]:

$$B_{ir}^*(s) = \frac{1 - B_i^*(s)}{s \bar{x}_i}. \quad (6)$$

Consider next the sojourn time for class 2. Let \tilde{y}_1 be the length of the class 1 busy period [23], \bar{y}_1 be its mean, and $C_1^*(s)$ be the Laplace transform of its pdf. Also let \tilde{x}_{2c} be

the length of a class 1 delay cycle with initial delay equal to the remaining service time of a class 2 packet seen by a class 1 arrival [20], \bar{x}_{2c} be its mean, and $B_{2c}^*(s)$ be the Laplace transform of its pdf. $S_2^*(s)$ is given by [22]:

$$S_2^*(s) = \frac{1 - \gamma_2}{1 + \gamma_1} \left(\frac{s + \lambda_1[1 - C_1^*(s)]}{s - \lambda_2 + \lambda_2 B_{2c}^*(s)} \right) B_2^*(s) \quad (7)$$

where $\gamma_1 = \lambda_1 \bar{y}_1$ and $\gamma_2 = \lambda_2 \bar{x}_{2c}$. $B_{2c}^*(s)$ can be obtained from results of an M/G/1 system with exceptional first service. It is given by [22]:

$$B_{2c}^*(s) = B_2^*(s + \lambda_1 - \lambda_1 C_1^*(s)) \quad (8)$$

where $C_1^*(s)$ satisfies the following equation:

$$C_1^*(s) = B_1^*(s + \lambda_1 - \lambda_1 C_1^*(s)). \quad (9)$$

3.1.3 Special Case – Exponential Service Time

For the M/G/1 model with two classes of priority, the sojourn time distribution is very difficult to compute. An exception is the model when the service time is exponentially distributed. Our approach is to first obtain analytic results for the case of exponential service times, and then use these results as the basis for selecting p . For exponential service times, we have

$$B_1^*(s) = B_{1r}^*(s) = \frac{\mu}{s + \mu} \quad (10)$$

and

$$B_2^*(s) = B_{2r}^*(s) = \frac{\mu}{s + \mu}. \quad (11)$$

Substituting Equations (10) and (11) into Equation (5), we get, for class 1:

$$S_1^*(s) = \left(\frac{1}{p} \right) \frac{\mu - p\lambda}{s + \mu - p\lambda} - \left(\frac{1 - p}{p} \right) \frac{\mu}{s + \mu}. \quad (12)$$

Inverting this Laplace transform, the cumulative distribution function of class 1 sojourn time is given by:

$$Prob [\tilde{s}_1 \leq x] = \frac{1 - p}{p} e^{-\mu x} - \frac{1}{p} e^{-(\mu - p\lambda)x} + 1. \quad (13)$$

For class 2 sojourn time, we need to first compute $C_1^*(s)$. Substituting Equation (10) into (9), we have

$$C_1^*(s) = \frac{\mu}{s + \lambda_1 - \lambda_1 C_1^*(s) + \mu}. \quad (14)$$

Solving this equation, we get

$$C_1^*(s) = \frac{s + \mu + \lambda_1 - \sqrt{(s + \mu + \lambda_1)^2 - 4\lambda_1\mu}}{2\lambda_1}. \quad (15)$$

We next derive $B_{2c}^*(s)$ for the case of exponential service time distribution. Substituting Equations (11) and (15) into Equation (8), and after simplification, we get

$$B_{2c}^*(s) = \frac{2\mu}{(s + \mu + \lambda_1) + \sqrt{(s + \mu + \lambda_1)^2 - 4\lambda_1\mu}}. \quad (16)$$

Taking its first derivative, and evaluating it at $s = 0$, we have $\bar{x}_{2c} = 1/(\mu - \lambda_1)$.

Finally, substituting Equations (11), (15), and (16) in Equation (7) and recognizing that γ_1 and γ_2 are given by $\lambda_1 \bar{y}_1$ and $\lambda_2 \bar{x}_{2c}$ respectively, we have, after some simplifications:

$$S_2^*(s) = \left(\frac{\mu - \lambda_1 - \lambda_2}{\mu + s} \right) \frac{(s + \lambda_1)\psi - 2\lambda_1\mu}{(s - \lambda_2)\psi + 2\lambda_2\mu}, \quad (17)$$

where

$$\psi = s + \mu + \lambda_1 + \sqrt{(s + \mu + \lambda_1)^2 - 4\lambda_1\mu}.$$

The inversion of this Laplace transform can be performed numerically using the Matlab program described in [24]. After this inversion, numerical results for the cumulative distribution function of class 2 sojourn time, *i.e.*, $Prob[\tilde{s}_2 \leq t]$, are obtained.

Once we are able to compute $Prob[\tilde{s}_1 \leq t]$ and $Prob[\tilde{s}_2 \leq t]$, the probability that a packet is delivered on-time, as given by Equation (4), can readily be computed.

3.1.4 Numerical Results

We now present numerical results for q_s , the percentage of packets that are delivered on-time as a function of p . The key input parameters are the traffic intensity ρ , which is given by $\rho_1 + \rho_2$, and the pdf $f_{\tilde{d}}(t)$ of the T/H value of a packet. In our numerical examples, μ is equal to 1, and λ is selected such that the values of ρ considered are 0.1, 0.3, 0.5, 0.7, and 0.9. In an attempt to study different scenarios of packet deadlines, the following random variables are considered for the T/H value of arriving packets:

- (1) $2 + \tilde{e}$: \tilde{e} is exponentially distributed with mean x , x varies from 1 to 4
- (2) \tilde{u} : \tilde{u} is uniformly distributed between 2 and x , x varies from 4 to 10
- (3) $2 + \tilde{a}$: \tilde{a} is normally distributed with mean x and standard deviation $x/3$,
 x varies from 1 to 4

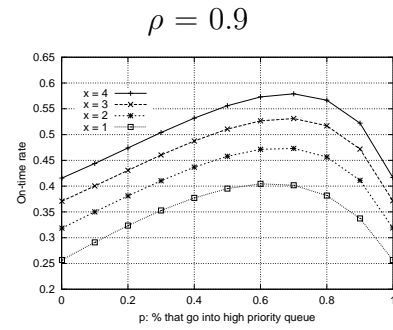
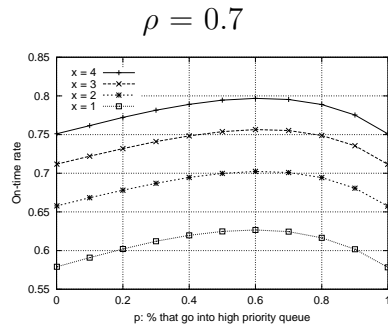
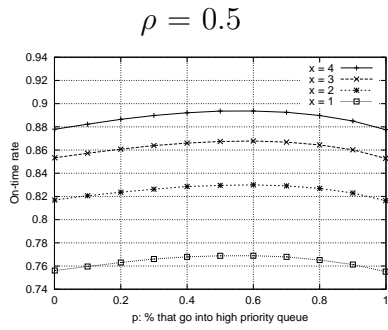
For the normal distributions, the standard deviation is selected to be 1/3 of the mean. This selection will result in a 99.7% probability that the random variable \tilde{a} is non-negative¹. Distributions (1) and (2) are defined such that the minimum and maximum T/H values are 2 and 6 respectively. Since $\mu = 1$, these T/H values correspond to 2 and 6 times the mean service time. As to distribution (3), the probability that T/H value is less than 2 or larger than 6 is very small (0.3% in either case).

It was found that when $\rho = 0.1$ and 0.3, the value of p has very little impact on performance. This is due to the fact that the load on the system is light, and the waiting time in queue is not significant. As ρ increases, the impact of p becomes more noticeable. In Figure 3, q_s is plotted against p for $\rho = 0.5, 0.7,$ and 0.9. There are three sets of graphs, corresponding to the three T/H distributions considered. Within each graph, results for different values of x are plotted.

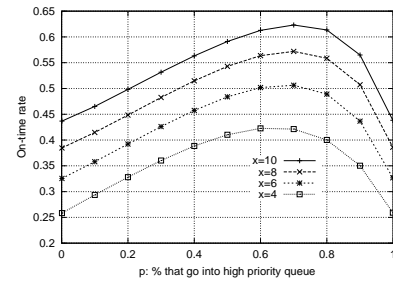
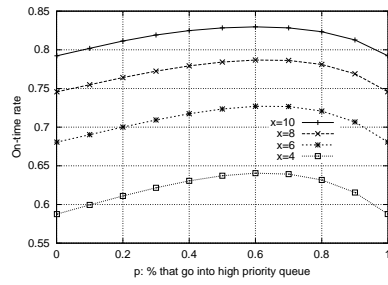
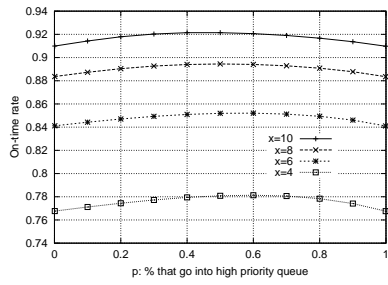
Let p_m be the value of p that yielded the highest percentage of on-time packets. For the cases considered, we observe that p_m ranges from 0.5 to 0.7 for $\rho \leq 0.7$ and from 0.6 to 0.7 when $\rho = 0.9$. Within each range, the actual performance is not very sensitive to p . We also observe that the performance degrades quickly when $p > 0.8$. The same observations were also made when $p < 0.4$. We thus conclude that a good value of p is within the range [0.5, 0.7].

In general, real-time traffic may have complex service time requirements. The assumption of exponential service time distribution is made for mathematical tractability. However, as can be seen in Section 4, where a network model is considered and more realistic traffic models are used, our approach of selecting p , which is based on analytic results for exponential service times, yields good performance in terms of on-time delivery. Furthermore, real-time traffic may have diverse deadline requirements. In Section 4, the notion of a deadline parameter

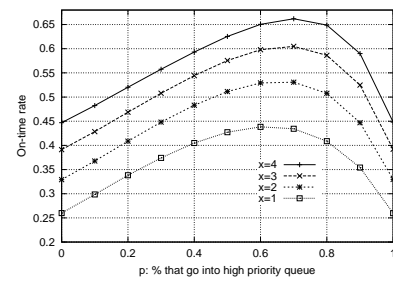
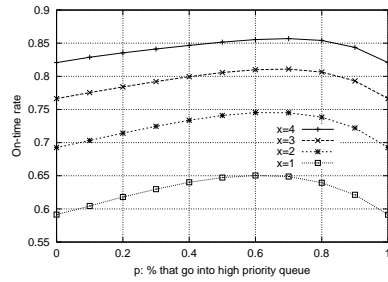
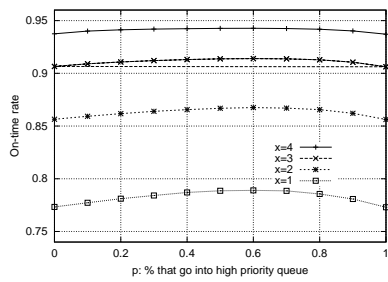
¹This percentage is based on the 68-95-99.7 rule.



Distribution (1)



Distribution (2)



Distribution (3)

Figure 3: Selection of p

is introduced, and more versatile deadline requirements will be considered when we use simulation to evaluate the merit of the T/H- p algorithm.

3.2 Implementation of T/H- p

In this section, an implementation of the T/H- p algorithm, with time complexity $O(1)$, is presented. Our implementation is based on the use of a threshold L , which is adjusted at regular intervals such that the fraction of packets that are sent to the high priority queue is p . These intervals are referred to as measurement intervals.

During each measurement interval, data for the following variables are collected:

V_{min}	=	minimum T/H value observed
V_{max}	=	maximum T/H value observed
n_h	=	number of packets entered the high priority queue
n_t	=	total number of packets observed

We now describe our implementation. At initialization, V_{min} is set to a large value, V_{max} , n_h and n_t are set to 0, and L can take any positive value. Each time a packet enters the system, its T/H value is calculated. Let this value be v . The following steps are performed:

```

 $n_t$  is incremented by 1
if ( $v < V_{min}$ )  $V_{min} = v$ 
if ( $v > V_{max}$ )  $V_{max} = v$ 
if ( $v \leq L$ ) send packet to high priority queue,  $n_h$  is incremented by 1
else send packet to medium priority queue

```

At the end of each measurement interval, the range of T/H values observed, as given by $[V_{min}, V_{max}]$, is compared with the threshold L . There are three cases: (a) $V_{max} \leq L$, (b) $L \leq V_{min}$, and (c) $V_{min} < L < V_{max}$. For cases (a) and (b), L is outside the range. The new value of L (denoted by L') is obtained by interpolation, *i.e.*, $L' = V_{min} + p \times (V_{max} - V_{min})$.

For case (c), L is within the range. Our approach to determining L' is as follows. During the measurement interval, the total number of packets observed is n_t . Ideally, the number of packets entering the high priority queue is $p \times n_t$. However, the actual number is n_h . If

$p \times n_t < n_h$, the value of L used during the measurement interval was too large. We again use interpolation and determine L' using the following equation (see Figure 4 (a)):

$$\frac{L - L'}{L - V_{min}} = \frac{n_h - p \times n_t}{n_h}. \quad (18)$$

Solving this equation, we get

$$L' = V_{min} + (L - V_{min}) \frac{p \times n_t}{n_h}. \quad (19)$$

On the other hand, if $p \times n_t \geq n_h$, the value of L used during the measurement interval was

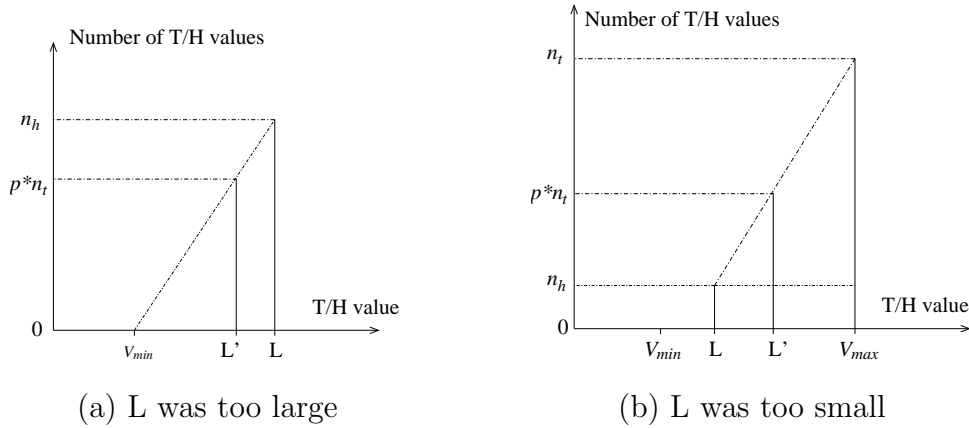


Figure 4: L adjustment in T/H-p

too small. Using interpolation, we have (see Figure 4 (b)):

$$\frac{V_{max} - L'}{V_{max} - L} = \frac{n_t - p \times n_t}{n_t - n_h}. \quad (20)$$

L' is thus given by:

$$L' = L + (V_{max} - L) \times \frac{p \times n_t - n_h}{n_t - n_h}. \quad (21)$$

This concludes the description of our implementation. The length of the measurement interval is a tunable parameter. In general, it should be much longer than the mean service time so as to obtain a sufficiently large sample of T/H values.

As mentioned earlier, our implementation of T/H-p has constant time complexity. This can be explained as follows. Upon packet arrival, V_{min} , V_{max} , n_t , and n_h are updated where appropriate. None of these operations depend on the queue length. The adjustment of L occurs at a time scale much longer than average packet service time. The operations required are also not affected by the queue length.

4 Performance of T/H- p

In this section, the effectiveness of T/H- p is evaluated by simulation. A custom packet-level simulator was developed for this purpose. In our investigation, a value of 0.5 will be used for p . This value is within the range of 0.5 and 0.7 discussed in the last section. It yields good performance, but a slightly higher value of p may perform better under certain traffic conditions. So the simulation results obtained should be viewed as conservative estimates of the performance of the algorithm.

4.1 Simulation Model

Consider first the modeling of ADU delivery from a given sender to a given receiver. This is depicted in Figure 5. ADUs are generated at the sender. Each ADU is characterized by five

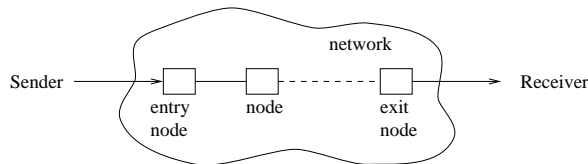


Figure 5: Concatenation from sender to receiver

attributes: ADU size, sender and receiver addresses, deadline and arrival time. Segmentation of an ADU into packets is performed at the sender before the packets are admitted to the entry node. For convenience, the processing time at the sender to prepare the packets for transmission is not included in the model. This assumption does not affect the conclusion on the merit of deadline-based scheduling.

Packets are routed through the network until they reach their exit node. They are then delivered to the receiver where packets belonging to the same ADU are reassembled. Again, for convenience, the processing time at the destination is not included. It is also assumed that fixed shortest-path routing is used to route packets from sender to receiver, routers have infinite buffer size, there is no transmission error, and packet transmissions are not delayed by any flow or congestion control mechanisms.

For a real-time ADU, its network-layer packet deadline is modeled as follows. Let x_a be

the *end-to-end latency* within the network when there is no queuing and no segmentation. Also let x_p be the end-to-end propagation delay, z the size of the ADU, and c_j the capacity of the j -th channel along the path based on shortest-path routing. Then x_a can be estimated by $x_a = x_p + \sum_j \frac{z}{c_j}$. The allowable delay within the network is assumed to be proportional to x_a . It follows that the deadline of all packets of the ADU is given by $d = \text{arrival time} + kx_a$, where k is a multiplier referred to as the *deadline parameter*. In general, a smaller k means that the ADU has a more urgent deadline. By considering a range of values of k , we can test the effectiveness of T/H- p under various degrees of urgency in ADU delivery.

Finally, the clocks at the sender, receiver, and routers along the path are assumed to be perfectly synchronized.

4.2 Simulation Results

In choosing the network topology for our simulation experiments, we take into consideration factors such as connectivity, size, and geographical coverage. The selected topology is shown in Figure 6; it represents a backbone network similar to, but slightly more complex than, the Abilene topology [25].

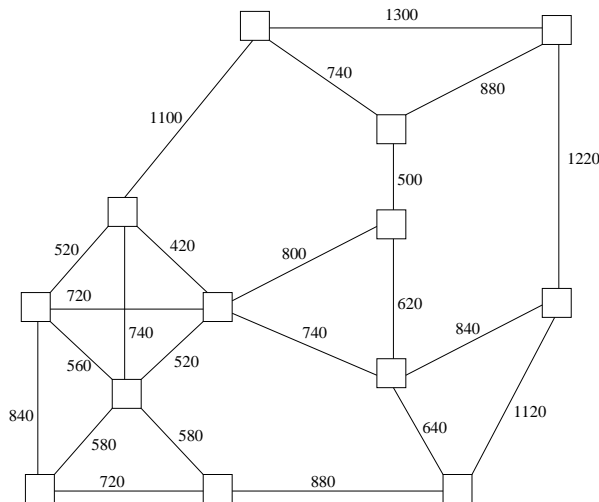


Figure 6: Network model

The value shown on each link is the distance in miles. This information is used to

determine the propagation delay². Routers are interconnected by full-duplex channels. The capacity of each channel is assumed to be 155 Mbit/sec. Albeit uniform channel capacity, due to shortest-path routing and the traffic model used, different channels may carry different load.

ADUs may be generated by applications from any node in the network. It is assumed that the ADU interarrival time is exponentially distributed, and the aggregated arrival rate is λ , in number of ADUs per second. For each arriving ADU, the entry node and exit node are selected at random. The size of each ADU is assumed to belong to one of two ranges: [500, 1500], and [1500, 500000], in bytes. The first range reflects small ADUs, *i.e.*, one packet per ADU. 25% of ADUs generated are small ADUs. ADU size is assumed to be uniformly distributed within each of these two ranges. In our investigation, a maximum packet size of 1500 bytes is assumed; this is the maximum Ethernet frame size and has been widely used on the Internet.

We first performed a simulation experiment to determine a proper value for the length of the measurement interval used in the implementation of T/H- p (denoted by I). In our experiment, $\lambda = 1200$ and the deadline parameter is assumed to be given by: $k = 1 + \tilde{e}_1$, where \tilde{e}_1 is exponentially distributed with mean 0.5. For convenience, all ADUs are of type real-time; this assumption is not expected to have a significant impact on our conclusion. I values in the range of 0.05 to 2.5 seconds were tested, and our results indicate that the on-time performance is not sensitive to I . Since too small a I may be costly in terms of overhead, and too large a I may be too slow in reacting to changing traffic conditions, we conclude that $I = 1$ second is a reasonable choice for the measurement interval.

Effect of ADU Arrival Rate We first study the effect of ADU arrival rate λ on performance. In our experiments, all ADUs are of type real-time. This would place the heaviest demand on the network resources. Six distributions of deadline parameter k are considered. They are:

- (1) $k = 1 + \tilde{e}_1$: \tilde{e}_1 is exponentially distributed with mean 0.5

²A propagation delay of one millisecond per 120 miles is used [26].

- (2) $k = \tilde{u}_1$: \tilde{u}_1 is uniformly distributed between 1 and 2
- (3) $k = 1 + \tilde{e}_2$: \tilde{e}_2 is normally distributed with mean 0.5 and standard deviation 0.166
- (4) $k = 2 + \tilde{e}_3$: \tilde{e}_3 is exponentially distributed with mean 1
- (5) $k = \tilde{u}_2$: \tilde{u}_2 is uniformly distributed between 2 and 4
- (6) $k = 2 + \tilde{e}_4$: \tilde{e}_4 is normally distributed with mean 1 and standard deviation 0.333

For the normal distributions, the standard deviation is chosen to be 1/3 of the mean. As a result, random variates generated for these distributions will be non-negative 99.7% of the time. In our simulation experiments, random variates with value less than zero are discarded.

In Figure 7, the percentage of ADUs that are delivered on-time under T/H, T/H- p and FCFS is plotted against the ADU arrival rate λ . For all three algorithms, results for both

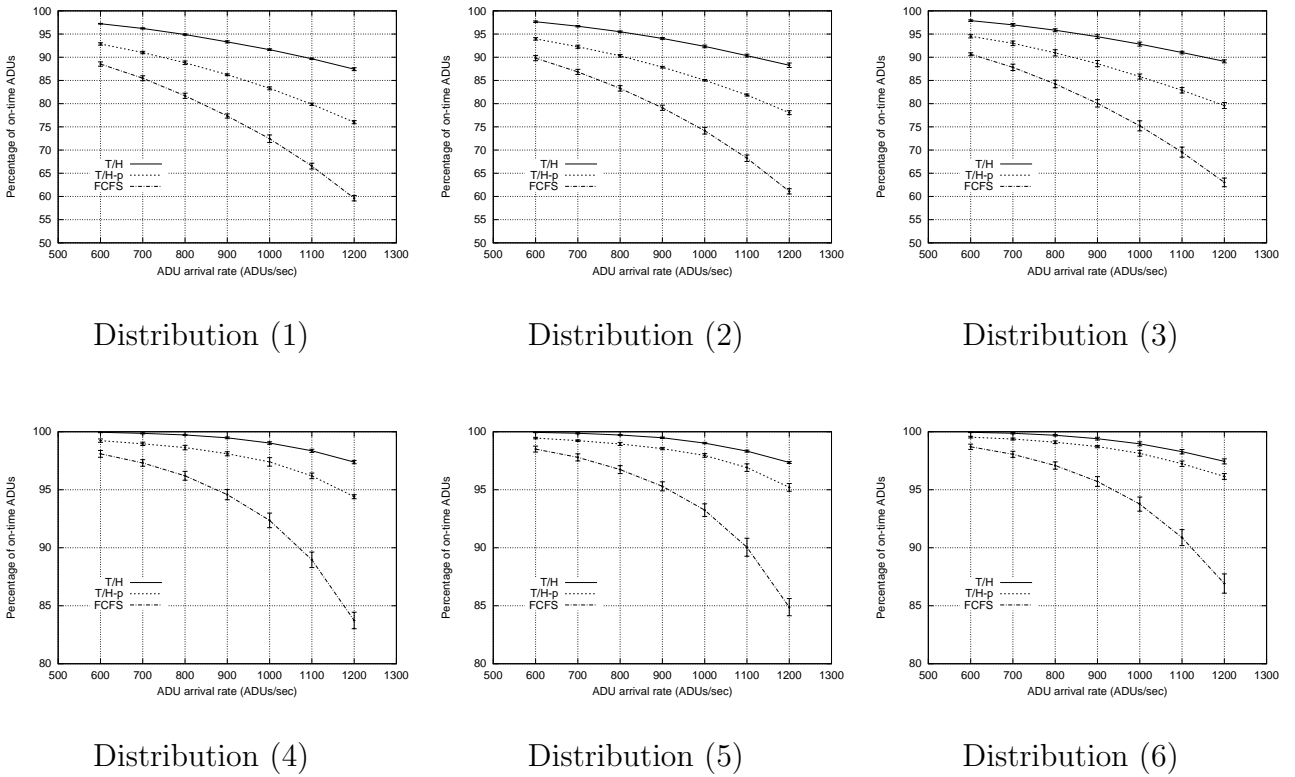
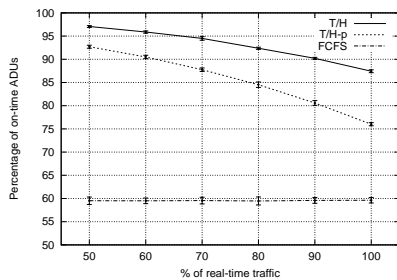


Figure 7: Effect of ADU arrival rate

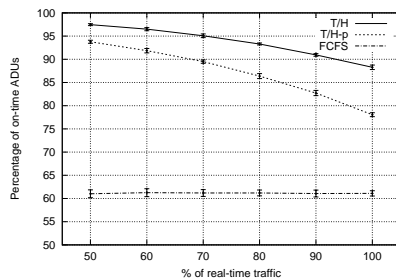
the mean and 99% confidence interval are shown. As expected, the performance of all three algorithms (T/H, T/H- p , and FCFS) degrades as λ increases. Among these algorithms, T/H-

p results in much better performance than FCFS, but is not as good as T/H. The performance difference of the three algorithms becomes more significant when the load is heavier. Our results also show that the percentage of ADUs that are delivered on-time is improved when the deadline is less stringent (see, *e.g.*, Distributions (4) vs. (1)). The performance difference of the three algorithms becomes smaller as well. In addition, for deadline distributions with the same mean (Distributions (1),(2),(3), and Distributions (4),(5),(6)), the performance of all three algorithms is not very sensitive to the actual distribution. The above results indicate that T/H- p is effective in delivering real-time documents.

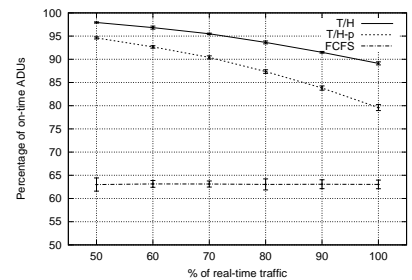
Effect of Percentage of Real-time Traffic The next experiment is designed to study the impact of the percentage of real-time traffic r on performance. For this experiment, the ADU arrival rate is assumed to be 1200 ADUs/sec. The same six distributions of k are considered. In Figure 8, the percentage of real-time ADUs that are delivered on-time is



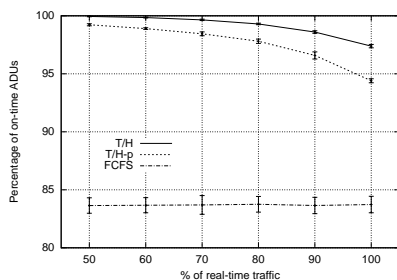
Distribution (1)



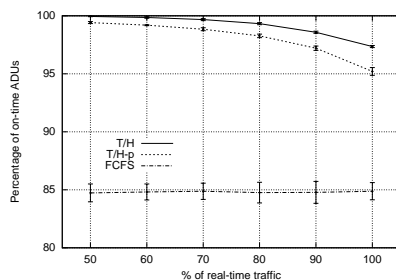
Distribution (2)



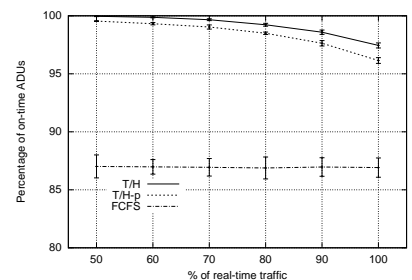
Distribution (3)



Distribution (4)



Distribution (5)



Distribution (6)

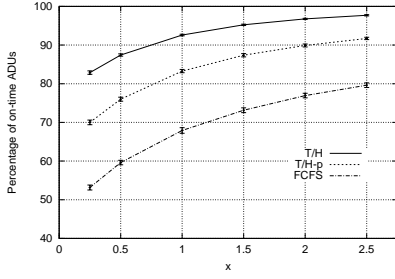
Figure 8: Effect of percentage of real-time traffic

plotted against r . It can again be observed that T/H- p is superior to FCFS, but not as good as T/H. Note that the performance of FCFS is not affected by the percentage of real-time traffic. This is because it does not give preferential treatment to any traffic type: real-time or best-effort. As r increases, the level of real-time traffic becomes higher, resulting in a lower percentage of real-time traffic being delivered on-time. However, the performance of T/H and T/H- p are still much better than FCFS even when $r = 100\%$. These results again demonstrate the effectiveness of T/H- p in delivering real-time documents.

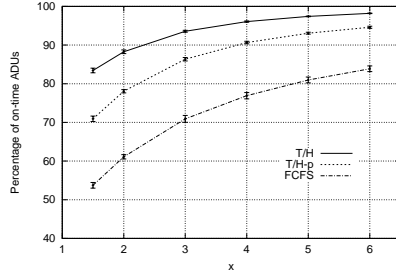
Effect of Deadline Parameter We next study the performance of T/H- p as a function of the deadline parameter k . In general, a larger value of k means a longer time before the deadline expires. The distributions of k considered are the same as those in the last two experiments except that for each distribution, a range of values of k are considered. These distributions are:

- (1) $k = 1 + \tilde{e}_5$: \tilde{e}_5 is exponentially distributed with mean x , x varies from 0.25 to 2.5
- (2) $k = \tilde{u}_3$: \tilde{u}_3 is uniformly distributed between 1 and x , x varies from 1.5 to 6
- (3) $k = 1 + \tilde{e}_6$: \tilde{e}_6 is normally distributed with mean x and standard deviation $x/3$,
 x varies from 0.25 to 2.5
- (4) $k = 2 + \tilde{e}_7$: \tilde{e}_7 is exponentially distributed with mean x , x varies from 0.5 to 5
- (5) $k = \tilde{u}_4$: \tilde{u}_4 is uniformly distributed between 2 and x , x varies from 3 to 12
- (6) $k = 2 + \tilde{e}_8$: \tilde{e}_8 is normally distributed with mean x and standard deviation $x/3$,
 x varies from 0.5 to 5

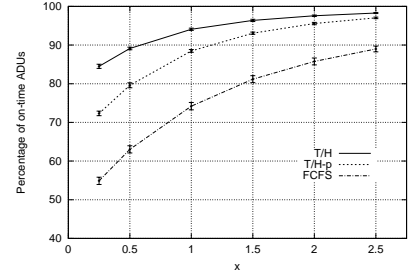
The above distributions are defined such that a smaller value of x means a tighter deadline. In our experiments, the ADU arrival rate $\lambda = 1200$ ADUs/sec, and all ADUs are of type real-time. In Figure 9, we show the percentage of ADUs that are delivered on-time as a function of the parameter x , for the six distributions of k considered. We observe once again that T/H- p performs significantly better than FCFS, but is not as good as T/H. We also observe that as the deadline becomes less stringent (or x becomes larger), a larger fraction of ADUs are delivered on-time. On the other hand, for more stringent deadlines, the performance advantage of T/H or T/H- p over FCFS is more clearly shown.



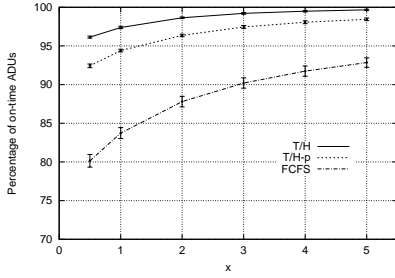
Distribution (1)



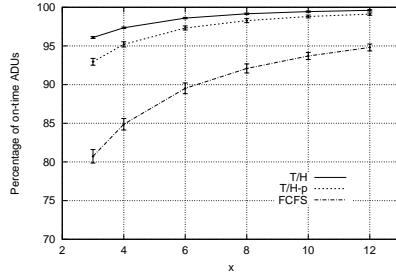
Distribution (2)



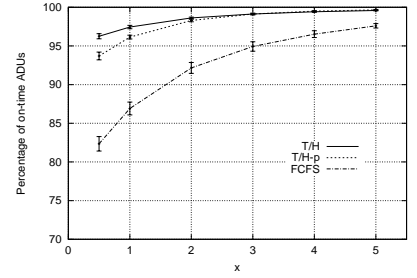
Distribution (3)



Distribution (4)



Distribution (5)

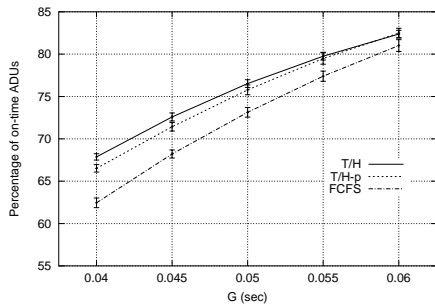


Distribution (6)

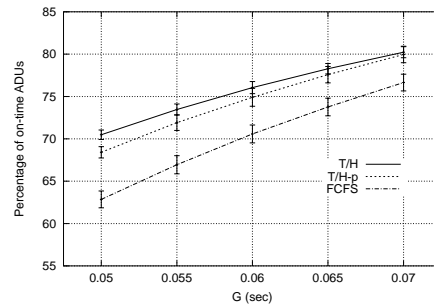
Figure 9: Effect of deadline parameter

In addition, we have performed simulation experiments for the case where the difference between the delivery deadline and arrival time is the same for all packets. Let G be this difference. In Figure 10, the on-time performance is plotted against G for ADU arrival rates λ equal to 1000 and 1200 ADUs/sec. For $\lambda = 1000$, the values of G considered vary from 40 ms to 60 ms. These values are higher than the mean end-to-end latency for our example network, which is approximately 33 ms. For $\lambda = 1200$, the values of G considered vary from 50 ms and 70 ms (the corresponding mean end-to-end latency is approximately 45 ms). We again observe that both T/H and T/H- p outperform FCFS by a significant margin, and that the performance difference of these three algorithms becomes smaller when the deadline is less stringent.

We have also considered the case where the deadline parameter k is a constant. This means that for all packets routed along the same end-to-end path, the difference between the delivery deadline and the arrival time is the same and proportional to the ADU end-to-end



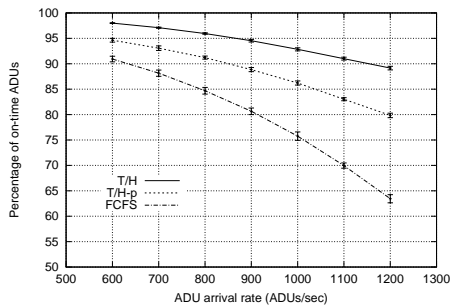
$$\lambda = 1000$$



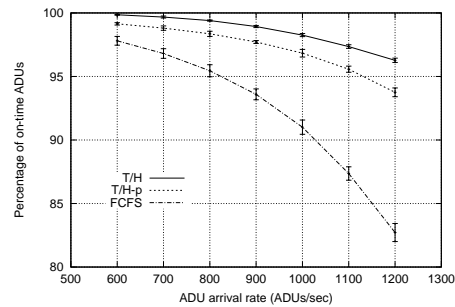
$$\lambda = 1200$$

Figure 10: Constant end-to-end deadline

latency. The values of k considered are 1.5 and 2.5. Results for the on-time performance are plotted in Figure 11. Once again, these results show that T/H-p is superior to FCFS and



$$k = 1.5$$



$$k = 2.5$$

Figure 11: Constant deadline parameter

that a less stringent deadline leads to smaller performance difference between T/H, T/H-p, and FCFS.

In conclusion, the results in this section confirm that T/H and T/H-p are effective algorithms in support of the delivery of real-time documents in a network environment. Their performance is significantly better than that of FCFS, especially when the ADU arrival rate increases, when the deadline becomes more stringent; or when the fraction of real-time traffic is decreased. Between these two algorithms, T/H has better performance.

The performance results presented in this section are based on the network example shown in Figure 6. One may be interested in the impact of network topology on performance. In our investigation, we observed that on-time performance is affected by the end-to-end delay experienced by an ADU, which is in turn affected by the utilization of the channels along the path from source to destination. Our example network has 13 nodes and 23 channels. For a given traffic level, the utilization of the various channels may be quite different. For example, when λ equals 1200 ADUs/sec, the channel utilization under shortest path routing ranges from 7% to 90%, even though the capacity of each channel is assumed to be the same and equal to 155 Mbps. We expect that when a different network topology (with possibly heterogeneous channel capacities) is used, there will still be a range of values for the channel utilizations, and as a result, the new topology will not have a significant impact on the performance difference of T/H, T/H- p and FCFS.

4.3 Additional Performance Results for T/H- p

In this section, we discuss additional simulation results that we have obtained for the performance of T/H, T/H- p and FCFS. These include results for finite router queue length; continuous-media data delivery; large networks where estimated hop count information is used; and a multi-service scenario where some but not all routers along the data path employ deadline-based scheduling [27].

Effect of Finite Router Queue Length The performance results presented in Section 4.2 assume infinite queue length at the routers. To study the impact of queue length on performance, we repeated the experiments in Figure 7 for deadline distribution (1) where each outgoing port has a fixed amount of buffer. Two buffer sizes are considered: 3.1 Mbyte and 775 Kbyte, corresponding to 160 ms' and 40 ms' worth of data at a link capacity of 155 Mbit/sec. The amount of buffer space used by real-time traffic and best effort traffic can be at most 90% and 10%, respectively.

The percentage of ADUs that are delivered on-time is shown in Figure 12. We observe that for the two buffer sizes considered, the performance of T/H- p is significantly better

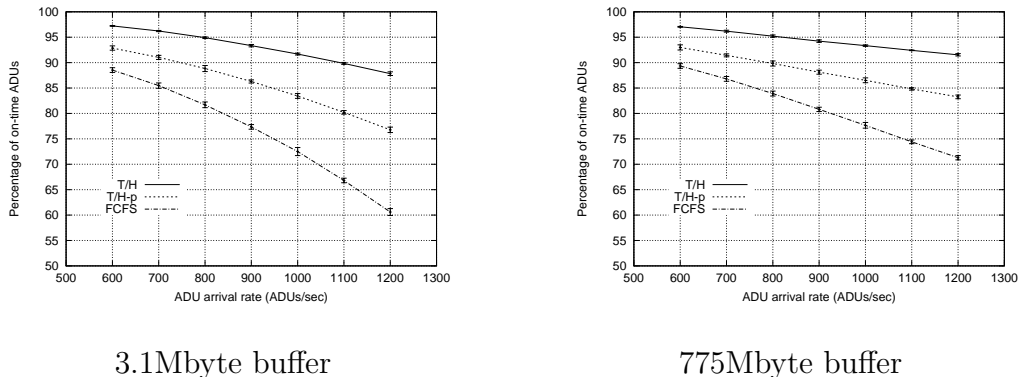


Figure 12: Effect of finite queue length

than FCFS, although it is not as good as T/H. Note that with finite buffers, packet loss may occur when the load becomes heavy. For the results shown in Figure 12, the highest observed loss rates are 0.8% for a 3.1 Mbyte buffer, and 9% when the buffer size is 775 Kbyte. We recognize that it is not a good idea to operate a network with high loss rates; nevertheless, our results show that T/H- p is superior to FCFS.

Continuous-Media Data We next discuss our results on the effectiveness of deadline-based scheduling for the transmission of video data. In our analytical model presented in Section 3.1, Poisson arrival of ADUs was assumed so that the analysis is tractable. This may not be a good assumption for video traffic. We therefore used video traces that are available from public domain as input to our simulation to evaluate the performance of T/H, T/H- p , and FCFS when video streams share the network resources with background non-video traffic [27]. In video transmission, each video frame is considered an ADU. Because of the isochronous nature of video playback procedure, a constant end-to-end deadline called *video playback delay* is used for all frames of a video stream.

Let λ be the arrival rate of video frames. In Figure 13 (a), we plot the video frame on-time rate against λ for a playback delay of 40 ms for the network model described in Section 4.2. We also show in Figure 13 (b) the video frame on-time rate when the video playback delay is varied. We observe from the results in these two figures that T/H and T/H- p have very similar performance. Both are superior to FCFS, especially when the load

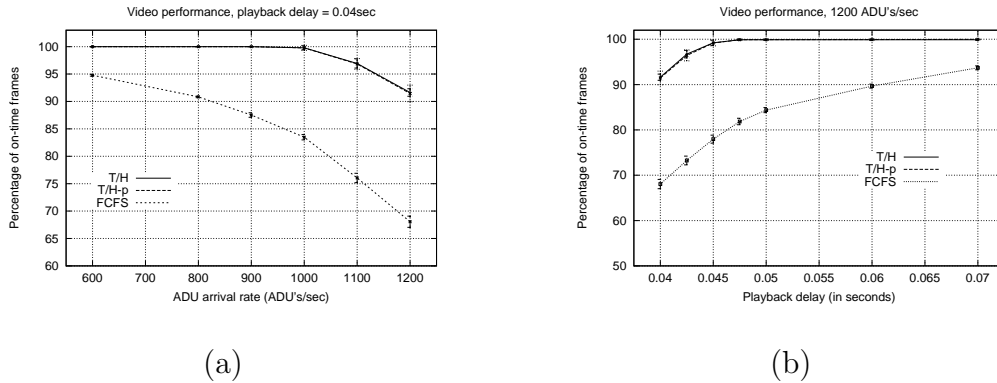


Figure 13: Video transmission performance

is heavy or the playback delay is small. In addition, we have investigated the impact of the degree of burstiness of a video stream on the performance of T/H- p and found that the impact is not significant.

Large Networks and Estimated Hop Count In a large network that consists of multiple autonomous systems (AS's), accurate hop count information may not always be available because the various AS's traversed by a packet may be administered by different network service providers. We have evaluated the performance of T/H, T/H- p and FCFS when only an estimate of the total number of hops for packets that traverse multiple AS's is available. In our experiments, we assume that each router has knowledge of the hop count information within its local AS, the set of transit AS's traversed by a packet, and estimates of hop count at each transit AS and at the destination AS. Our results again showed that the performance of T/H and T/H- p is superior to FCFS in terms of the percentage of ADUs that are delivered on-time. Furthermore, for both algorithms, using a hop count estimate ranging from 50% to 250% of the actual hop count at a transit or destination AS would not have a significant impact on performance. We recognize that in practice, it is difficult to determine the hop count. However, our results indicate that only a rough estimate is needed.

Multi-Service Scenario Our investigation so far has been based on network configurations where deadline-based scheduling is used in every router in the network. Another

configuration of interest is a multi-service scenario where some routers in a network use deadline-based channel scheduling, while others use FCFS. We have evaluated the performance of such a multi-service scenario using simulation. For the network model depicted in Figure 6, we vary the number of routers that employ deadline-based scheduling (denoted by M) from 0 to 13; these two values correspond to the use of FCFS and deadline-based scheduling in every router respectively. In Figure 14, the percentage of ADU's that are de-

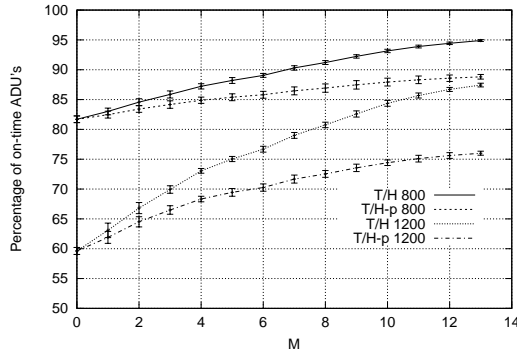


Figure 14: Performance results for multi-service scenarios

livered on-time is plotted against M for $\lambda = 800$ and 1200 . We observed that the on-time performance improves as the number of routers that employ deadline-based scheduling increases. With respect to the placement of such routers, we found that the performance gains are more significant if deadline-based scheduling is used in more heavily utilized channels.

In summary, the additional results in this section further support the merit of T/H- p in providing effective support to real-time data delivery.

5 Practicality of T/H- p in High-Speed Routers

In this section, we discuss the practicality of T/H and T/H- p in terms of their implementation in high-speed routers. T/H has time complexity $O(\log N)$. It was recognized that “software solutions, which are logarithmic in time complexity, are typically not fast enough to keep up with the packet transmission rate” [12]. Thus hardware solutions are required. There are a number of hardware priority queue architectures that are reported in the literature. These include binary tree of comparators, FIFO priority queue (one FIFO queue per priority),

shift register priority queue, systolic array priority queue, a hybrid shift register and systolic array [12], and pipelined heap [13, 14]. These architectures require hardware resources that increase with the number of priority levels, or in the context of T/H, the number of dissimilar T/H values associated with packets to be processed.

In comparison, T/H- p employs a small and constant number of FIFO queues per outgoing port. Its $O(1)$ time complexity allows for software implementations. Efficient implementations in hardware are also possible. Specifically, FIFO queues are easy to implement. Each FIFO only needs two pointers for the head and tail of the queue [28], and the implementation is not affected by the aforementioned issue related to the number of dissimilar T/H values. Since the number of priority levels is small, the hardware cost is not significant. T/H- p requires data collection for V_{min} , V_{max} , n_h and n_t . Each time a packet is processed, one or more of these data items need to be updated. Since the operations involved are compare, store, and increment, these updates can be done easily in hardware. The algorithm to adjust L is also straightforward. Based on the above discussions, T/H- p is an attractive candidate for high-speed routers when both performance and implementation cost are taken into consideration.

6 Extension to More Than 2 Queues

In T/H- p , real-time packets are sent to two different queues, depending on their T/H values. This idea can be generalized to more than 2 queues. It would provide more differentiation of packet urgency and should result in improved performance. In this section, we study the performance advantage that may be achieved by such an extension, referred to as T/H- $p(m)$, where m is the number of queues. Note that T/H- $p(m)$ becomes T/H as m goes to infinity.

6.1 The T/H- $p(m)$ Algorithm

The T/H- $p(m)$ algorithm is depicted in Figure 15. The m priority queues for real-time packets are numbered such that Q1 has the highest priority and Q m has the lowest priority. Accordingly, the T/H values of packets in Q1 are the smallest among all queues, followed by

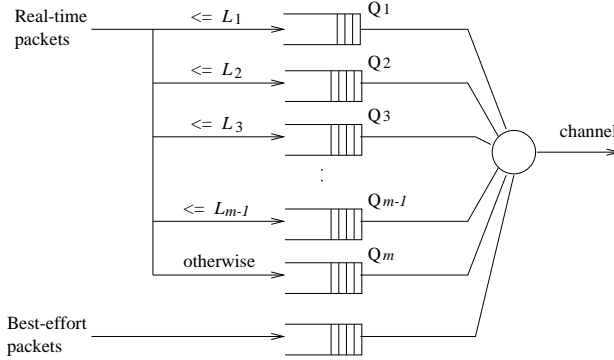


Figure 15: Queueing model for $T/H-p(m)$

Q2 which has the next smallest, then Q3, Q4, until Qm. The fraction of real-time packets that are sent to Q_i is p_i , $i = 1, 2, \dots, m$, where $\sum_{i=1}^m p_i = 1$. Head-of-the-line priority is used to serve packets in these queues, including the best-effort queue, which has lower priority than Q_m . Our implementation of $T/H-p$ can readily be extended to implement $T/H-p(m)$. The details can be found in [27].

As to the algorithm complexity, $T/H-p(m)$ is a constant time algorithm. This is due to the fact that the number of queues is constant. At packet arrival, a division is performed to calculate the packet's T/H value, the number of Boolean operations to select a queue for the incoming packet is $O(1)$, and a constant number of integer counters are incremented for data collection purposes. At packet departure, at most $m + 1$ queues are scanned to find a non-empty queue. The thresholds L_i , $1 \leq i \leq m$, are adjusted at end of each measurement interval only, and the amount of computation required is of constant time also.

6.2 Performance of $T/H-p(m)$

The simulation experiments reported in Section 4.2 are extended to include $T/H-p(m)$. Note that $T/H-p(m)$ and $T/H-p$ are the same when $m = 2$. For $m > 2$, we have not carried out any analysis or simulation to determine good values for p_i , $1 \leq i \leq m$. As a heuristic, we assume that the p_i has the same value for all i .

In Figure 16, the percentage of ADUs that are delivered on-time is plotted against m for $\lambda = 800$ and 1200 ADUs per second. Note that the performance of T/H and FCFS is not affected by m . As to $T/H-p(m)$, improved performance is observed when a larger value of m

is used. It is also observed that significant gain is realized when m is increased from 2 to 4. As m is further increased, the rate of improvement decreases with m , and the performance of $T/H-p(m)$ approaches that of T/H . $m = 4$ will be used in subsequent experiments.

We next study the effect of (i) ADU arrival rate, (ii) percentage of real-time traffic, and (iii) deadline parameter, on performance. The results are shown in Figure 17. For (i) and

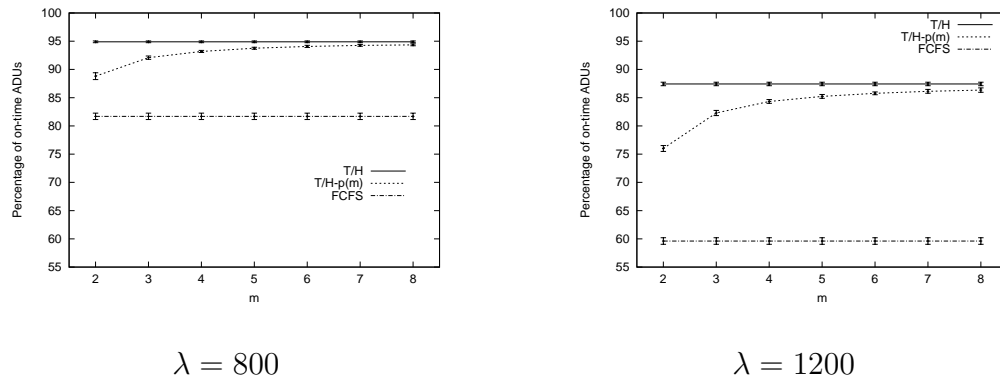


Figure 16: Varying m in $T/H-p(m)$

(ii), the deadline parameter k is assumed to be given by $1 + \tilde{e}_1$ where \tilde{e}_1 is exponentially distributed with mean 0.5. For (iii), k is assumed to be given by $1 + \tilde{e}_x$ where \tilde{e}_x is exponentially distributed with mean x where x varies from 0.25 to 0.5. In all cases, we observe

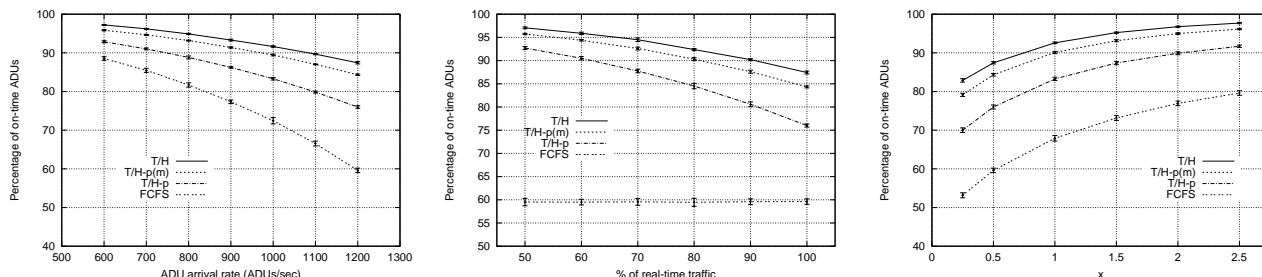


Figure 17: $T/H-p(m)$ performance

that $T/H-p(m)$ with $m = 4$ yields much better performance than $T/H-p$ (or $m = 2$). Its performance is also very close to that of T/H (or $m = \infty$). These results demonstrate that a small increase of m from 2 to 4 can lead to a noticeable improvement in performance.

7 Conclusion

A new deadline-based channel scheduling algorithm within our deadline-based network resource management framework, called T/H- p , has been developed. In this algorithm, a fraction p of packets that have smaller ratios of time until delivery deadline over number of hops left are given higher priority. The performance of T/H- p in terms of the percentage of ADUs that are delivered on-time has been evaluated by simulation. It was found that T/H- p is superior to FCFS, but is not as good as T/H. In terms of implementation in high-speed routers, T/H- p is less costly than T/H. Our conclusion is that T/H- p is an attractive candidate for high-speed routers when both performance and implementation cost are taken into consideration.

An extended version of T/H- p to $m > 2$ priority queues, called T/H- $p(m)$, has also been investigated. Efficient implementation of this extended algorithm is possible. Using simulation, we found that the performance of T/H- $p(m)$ improves with m and that a small value of m (*e.g.*, $m = 4$) would result in on-time performance close to that of T/H. These observations further support the merit of using T/H- p (or its extension) in high-speed routers.

Acknowledgment

This work was supported by the Natural Sciences and Engineering Council of Canada and by a grant from the Canadian Institute for Telecommunications Research under the NCE program of the Government of Canada.

References

- [1] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. *Internet-working: Research and Experience*, 1(1):3–26, 1990.
- [2] R. Geurin, S. Kamat, V. Peris, and R. Rajan. Scalable QoS provision through buffer management. In *Proc. of ACM Sigcomm'1998*, September 1998.
- [3] R. Braden, D. Clark, and S. Shenker. *Integrated services in the Internet architecture: an overview*, June 1994. Internet RFC 1633.
- [4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin. *Resource ReSerVation protocol (RSVP)*, September 1997. Internet RFC 2205.
- [5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. *An architecture for differentiated service*, December 1998. Internet RFC 2475.

- [6] J. W. Wong and Y. E. Liu. Deadline based network resource management. In *Proc. of ICCCN'2000*, pages 264–268, 2000.
- [7] Richard W. Conway, William L. Maxwell, and Louis W. Miller. *Theory of scheduling*. Addison Wesley, 1967.
- [8] J. Michael Moore. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science*, 15(1):102–109, September 1968.
- [9] J. Kobza and S. Liu. A head-of-line approximation to delay-dependent scheduling in integrated packet-switched networks. In *Proc. of IEEE Infocom'1989*, pages 1106–1113, 1989.
- [10] Henning G. Schulzrinne. *Reducing and characterizing packet loss for high-speed computer networks with real-time services*. PhD thesis, University of Massachusetts, 1993.
- [11] H. Schulzrinne, J. Kurose, and D. Towsley. An evaluation of scheduling mechanisms for providing best-effort, real-time communication in wide-area networks. In *Proc. of IEEE Infocom'1994*, pages 1352–1361, Toronto, Canada, June 1994.
- [12] Sung-Whan Moon, Jennifer Rexford, and Kang G. Shin. Scalable hardware priority queue architectures for high-speed packet switches. *IEEE Transactions on Computers*, 49(11):1215–1227, 2000.
- [13] R. Bhagwan and B. Lin. Fast and scalable priority queue architecture for high-speed network switches. In *Proc. of IEEE Infocom'2000*, March 2000.
- [14] A. Ioannou and M.G.H. Katevenis. Pipelined heap (priority queue) management for advanced scheduling in high-speed networks. *IEEE/ACM Transactions on Networking*, 15(2):450–461, April 2007.
- [15] Y. E. Liu and J. W. Wong. Deadline based channel scheduling. In *Proc. of the IEEE Globecom'2001*, pages 2358–2362, San Antonio, Texas, November 2001.
- [16] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of the ACM*, 20(1):46–61, January 1973.
- [17] M. Andrews. Probabilistic end-to-end delay bounds for earliest deadline first scheduling. In *Proc. of IEEE Infocom'2000*, pages 603–612, March 2000.
- [18] K. Zhu, Y. Zhuang, and Y. Viniotis. Achieving end-to-end delay bounds by EDF scheduling without traffic shaping. In *Proc. of IEEE Infocom'2001*, pages 1493–1501, 2001.
- [19] J. Liebeherr and D. E. Wrege. Priority queue schedulers with approximate sorting in output buffered switches. *IEEE Journal on Selected Areas in Communications*, 17(6):1127–1145, 1999.
- [20] Leonard Kleinrock. *Queueing systems, Vol. 2: Computer applications*. Wiley, New York, 1975.
- [21] D. D. Clark, S. Shenker, and L. Zhang. Supporting real-time applications in an integrated services packet networks: architecture and mechanism. In *Proc. of ACM Sigcomm'1992*, pages 14–26, August 1992.
- [22] John N. Daigle. *Queueing theory for telecommunications*. Addison Wesley, 1992.
- [23] Leonard Kleinrock. *Queueing systems, Vol. I: Theory*. Wiley, New York, 1975.
- [24] Lubomir Brancik. Programs for fast numerical inversion of Laplace transforms in MATLAB language environment, 1998. Brno University of Technology, Czech Republic.
- [25] Abilene backbone network. <http://abilene.internet2.edu/>.
- [26] J. Walrand and P. Varaiya. *High-performance communication networks*. Morgan Kaufmann, 2000.
- [27] Y. E. Liu. *Deadline based network resource management*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 2003.
- [28] S. Keshav and R. Sharma. Issues and trends in router design. *IEEE Communications Magazine*, 36(5):144–151, May 1998.