End-to-end Delay Analysis for Real-Time Distributed Networks *

Shu-Ngai Yeung
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213
Email: syeung@stat.cmu.edu

John Lehoczky
Department of Statistics
Carnegie Mellon University
Pittsburgh, PA 15213
Email: jpl@stat.cmu.edu

Abstract

This paper presents an application of real-time queueing network theory to a particular network which models a video-on-demand server. Packets from each stream have stochastic arrival patterns, stochastic computation times and end-to-end delay requirements. We derive closed from solution for the deadline miss rate of the packets. This methodology can be used to design admission control policies that provide statistical quality of service (QoS) guarantees. By considering the average computation requirements rather than worst-case requirements, the real-time queueing network approach significantly increases the server utilization. This paper makes a set of important contributions. First, it illustrates how real-time queueing network theory can be used to accurately predict the behavior of real-time systems in heavy traffic conditions. Second, it shows how one can calculate the fraction of tasks (packets) that will miss their end-to-end deadlines. Third, it presents new results on product form equilibrium distributions for multi-dimensional reflected Brownian motion processes when nodes are scheduled using EDF. Finally, it presents simulation results to illustrate the excellent accuracy of the real-time queueing network approach and how the methodology can be used to provide statistical QoS guarantees even when the worst-case utilization exceeds 100%.

1 Introduction

Hard real-time system scheduling theory focuses on ensuring that all tasks with hard deadlines meet those deadlines with certainty. The requirement that all tasks finish before their deadline elapses can only be met if the arrival pattern of tasks and their computational requirements are deterministic or deterministically bounded. That is, one assumes the task arrivals from one application are separated by a minimum interarrival time and their computational requirements are bounded above by their worst case computation times. The applications are, therefore, treated as periodic with a deterministic computation time [11]. Under this framework, hard real-time scheduling methods can be applied (such as the generalized rate monotonic or earliest deadline first scheduling algorithms), and the ability of the system to meet the deadlines can be determined using real-time scheduling theory.

This approach has been of major importance in the development of analytic methods for scheduling real-time systems; however, the worst case formulation that it requires can lead to substantial under-utilization of the system. Under certain conditions, the real-time scheduling algorithms can meet all task deadlines as long as the worst case processor utilization is less than 100%. There are, however, many cases in which the worst case utilization is substantially greater than the average case utilization. In such a situation, the actual processor utilization or throughput can be quite small. Consider, for example, a video-conferencing application. Figure 1 illustrates the bandwidth requirements that arise in three different types of scenes occurring in a video-conference: quiet, active and camera motion. The quiet scenes have very small bandwidth requirements, while the active scenes or instances of camera panning or zooming have much greater requirements. The accompanying histogram shows that the bandwidth requirements are usually very small, say 600kbp, but this may exceed 5,000kbp. Thus the worst case bandwidth requirements are roughly one order of magnitude larger than the average case. If one were to reserve bandwidth sufficient for the worst case, as much as 90% may be wasted.

The correctness conditions of real-time systems include timing correctness. While some real-time appli-
Recently, Bernat, Burns and Llamosí [3] defined the concept of *weakly hard real-time* systems in which a specified maximum number of tasks were permitted to miss their timing requirements in any window of specified size. This specifies a maximum deadline miss-rate and ensures that the misses cannot be clustered in any window. The analysis still requires worst case assumptions on the times between task arrivals and the computation times of each task. Consequently, worst case utilization must be held below 100%.

The worst case analysis methodology converts stochastic task arrivals and computations into a deterministic process from which real-time scheduling analysis can proceed. While this method can then guarantee task timing requirements, it can result in unrealistically low processor utilization. Most of the recent works on real-time scheduling have been done on improving system utilization while providing performance guarantees. Tia et. al. [14] provide probabilistic schedulability guarantees for semi-periodic tasks (with constant interarrival and variable execution times) under the Rate Monotonic (RM) scheduling algorithm. When the total maximum utilization of the processor is larger than one, the probability of a task missing its deadline is positive. Atlas and Bestavros [2] propose Statistical Rate Monotonic Scheduling (SRMS), which provide an acceptance test for semi-periodic tasks. All admitted jobs are guaranteed to meet their deadlines through a fixed priority assignment. The statistical QoS guarantees are in terms of the fraction of jobs admitted for each task. Abeni and Buttazzo [1] use a Constant Bandwidth Server (CBS) to perform probabilistic QoS guarantee for soft real-time tasks. The CBS is a service mechanism which integrates hard real-time and soft multimedia computing in a single system, under the EDF scheduling algorithm. In [1], Markovian queueing analysis is done for a semi-periodic task model and a generalization of the sporadic task model (tasks with variable interarrival time and constant computation time)

Figure 1: Sample of video-conferencing bandwidth requirements.

Figure 2: Histogram of bandwidth requirements for a video-conference.

cations (such as those arising in control systems) do require that deadlines be met under worst case conditions, there is, however, a large set of applications (including, for example, applications involving multimedia) in which some deadlines can be missed. In recent years, real-time scheduling theory has been expanded to cover a wider set of requirements including *firm deadlines* and *soft deadlines*. Firm deadlines can be missed, but there is no value in completing the task once the deadline has elapsed. Soft deadlines can be missed, but it is still valuable to complete them even if they are not too late. These two allow for less stringent enforcement of task timing requirements, but do not constrain the pattern of deadline misses. It is possible that deadline misses can cluster in which case the application would exhibit unacceptable behavior. More
theory develops queueing models for systems processing tasks that have timing requirements. The goal of this theory is to determine whether a given task set is schedulable, meaning that all tasks meet their deadlines in the hard deadline case, or to predict the fraction of tasks that will not meet their deadlines in the firm or soft deadline case. The theory should apply to a variety of scheduling (queue discipline) policies, especially those that are commonly used in real-time systems like EDF.

A real-time queueing network is, in principle, difficult to analyze, because one must keep track of the lead-time (time until the deadline elapses) for every task in the system. These task lead-times change dynamically with time, decreasing at rate 1. Thus, every task in the system has a lead-time which must be tracked. This means, in turn, that the dimensionality of a real-time queueing system is at least as large as the number of tasks active in the network. Hence the dimensionality is unbounded, a situation in which few analytic tools are available to analyze the behavior of such systems. Fortunately, when the load at each node of the network becomes large (the so-called “heavy-traffic” case) the lead-times of the tasks in the network become well-approximated by a deterministic function of the workloads at each of the nodes in the system. The mathematical foundations underlying this approximation have been fully developed by Doytchinov, Lehoczky and Shreve [3] for the single node case and Yeung and Lehoczky [15] for the feed-forward network case. The workloads at each of the nodes in the system can be studied using ordinary queueing network theory. In the heavy traffic case, the workloads converge (under certain conditions) to a limiting Brownian network, a drifted, reflected multi-dimensional Brownian motion restricted to an orthant, the dimensions of which corresponds to the number of nodes in the network. One can compute the equilibrium distribution of the workloads, either in closed form or numerically using the Brownian network theory [4]. Using real-time queueing theory, for each workload vector, one can approximate the vector of lead-times (lead-time profile) at each node. One could integrate some feature of interest of the lead-time profile at any workload state against the equilibrium workload distribution to compute the overall performance of that feature. For example, suppose the feature we are concerned with is whether any tasks have negative lead-times (i.e. they are missing their deadlines). We could consider all lead-time profiles having some negative components and integrate those with respect to the equilibrium workload distribution. The result would be an approximation of the fraction of tasks which miss their deadlines. While the theory applies only in heavy traffic, the heavy traffic case is the most important one for real-time systems, and in this paper we will see that these approximations are remarkably accurate. The most important conclusion that one can draw using this methodology is the determination of the number of applications creating streams of tasks that can be admitted into the system while still ensuring that the fraction of tasks which miss their deadline is kept to an acceptable level. For example, later in this paper, we determine the deadlines that tasks must have in order for the fraction of tasks that miss their deadline to be no greater than \(10^{-14}\). Moreover, simulation studies confirm the surprising accuracy of the real-time queueing theory predictions. While we present these results in the context of a relatively simple network, the theory is quite general.

One major advantage of applying heavy traffic theory is its insensitivity to the modeling assumptions. All that is needed is that external tasks arriving to the network do so according to a renewal process (with independent task interarrival times), and all tasks have independent computation requirements. The specific interarrival distribution and the computation time distributions at each node do not need to be known, just their means and coefficients of variation along with the route of the different tasks through the network. This information is sufficient to determine the multi-dimensional workload distribution. The lead-time profiles at each node depend upon the workloads at each of the nodes and the scheduling discipline at each node. The formulas for them are presented later in the paper.

This paper presents a series of new results. In particular it:

1. illustrates how real-time queueing theory can be used to accurately predict the behavior or real-time systems in heavy traffic conditions.
2. develops analytic formulas to determine the number of applications that can be supported by a network while ensuring that at most a specified fraction of tasks (packets) miss their end-to-end deadlines.
3. develops conditions under which the multi-dimensional reflected Brownian motion process representing the workload distribution will have an equilibrium distribution of product form. This is done without the assumption of a Jackson network, and it is done for the EDF queue discipline.
4. it presents simulation results to illustrate the ex-
cellent accuracy of the real-time queueing approach and how standard real-time scheduling algorithms can seriously underestimate the quantity of work that can be processed by these real-time systems.

This paper is organized as follows. In section 2 we present the basic model and notation. Section 3 presents the heavy traffic diffusion approximation for this model. This section also presents the conditions under which product form will hold. Section 4 presents an end-to-end delay analysis for the network, both for the EDF and for the FIFO scheduling policies. In section 5, we present simulation results to verify the accuracy of the theory. We use the analysis presented in section 4 to perform a delay analysis in section 6. The result of this analysis determines the number of applications that can be admitted which still ensuring the task deadline miss-rate is held below a specified bound. Section 7 presents our conclusions.

2 Basic Model

![Figure 3: Distributed network.](image)

We consider the simple network presented in Figure 3. The network consists of a single central node and $N$ local nodes. Each local node can support $K$ applications (e.g. video-conferencing sessions), each of which generates an input stream of tasks (packets). Here we use the words “application” and “session” interchangeably. We will often refer to the tasks from each stream as packets, although this need not be the case. Hence the total number of sessions in the system is $N \cdot K$. Session $k$, $1 \leq k \leq NK$ has a fixed, deterministic route through the network. All of the input streams arrive to the central node (indexed by 0). The streams are then transmitted to the appropriate second node, with streams $(j-1)K + 1$ to $jK$ being transmitted to node $j$ at which location final processing and display take place. The methods described can be applied to much more complex networks; however, the one presented here is straightforward to analyze and will help to illustrate the method.

We assume the external inter-arrival times of session $k$ packets form sequences of independent and identically distributed (i.i.d.) random variables with mean $1/\lambda_k$ and coefficient of variation (c.v.) $a_k$. Transmission times of session $k$ packets at node 0 are i.i.d. random variables with mean $m_{kA}$ and c.v. $b_{kA}$. The transmission times and computational requirements for session $k$ packets at the local node are i.i.d. with mean $m_{kB}$ and c.v. $b_{kB}$. The transmission times at the central node and at the local nodes are assumed to be independent. In general, the capacity of the transmission link at the central node is much higher than those at the local nodes. The traffic intensity associated with each session at the central node is defined by $\rho_k = \lambda_k m_{kA}$, while the total traffic intensity at the central node is $\rho_0 = \sum_{k=1}^{NK} \lambda_k m_{kA}$. The traffic intensity at the local node $j$ is $\rho_{jB} = \sum_{k=(j-1)K+1}^{jK} \lambda_k m_{kB}$.

Each of the session $k$ packets is assumed to have an end-to-end deadline assigned independently (within and across streams) from the cumulative distribution $G_k$. We assume packets waiting for transmission are stored in buffers with infinite capacity at each node. We must specify the scheduling policy at each of the nodes. In general, we will assume that the nodes use EDF or they use FIFO. The EDF queue discipline requires that each task (packet) has sufficient information about the deadline to ensure that the tasks can be processed in deadline order. It is interesting to note that for the above network, if all tasks arrive with constant and identical deadlines, then EDF and FIFO are identical. The network defined in Figure 3 is a good representative of video-on-demand (VoD) systems, in which video/audio data are streamed from the central video server to the customers’ TVs through local area switches. The highly variable arrival patterns and packet sizes of video/audio traffic gives rise to the stochastic assumptions in the queueing model described above. An important performance measure of a VoD system is the end-to-end delay of packets. In order to provide satisfactory QoS, the fraction of late packets during a session period must be bounded.

In this paper, $W_j(t)$, $0 \leq j \leq N$ will denote the node $k$ workload process (remaining transmission time and processing time unfinished in the queue) at time $t$. In the following discussion, if we omit the argument $t$, it should be understood that we refer to the stationary behavior of the associated process.
3 The Real-Time Queueing Network Approach

In this section, we develop the real-time queueing network approximation for the end-to-end delay of the packets in the various streams. To do this, we follow a three-step process for the heavy traffic case in which the traffic intensities at the various nodes in the network approach 1.

The first step assumes the workloads at each of the nodes in the network is given by \((W_0, W_1, \ldots, W_N)\). Given these workloads, one can use real-time queueing network theory to determine the lead-time profiles of each of the tasks (packets) at each node in the network. We do this for general assumptions on the initial deadlines associated with the packets in each stream and assume either EDF or FIFO processing at each node. The lead-time profiles can be used to determine the fraction of packets that have negative lead-times (i.e., are late). The result is a mapping taking the workloads at each node into the fraction of packets that are late. This mapping depends upon the arriving deadline distributions associated with each stream and the scheduling policy at each node. Real-time queueing theory was first developed by [8, 9] and formalized for the single node case by [5]. For feed-forward networks with EDF or FIFO scheduling, such as the network shown in Figure 3, the mathematical foundations underlying real-time queueing are presented in [15].

The second step is based on heavy traffic queueing network theory. In the heavy traffic case, under very general conditions (including general arrival processes and service time assumptions, multiple application types and general routing schemes) the workload and occupancy of all the nodes in the network can be well approximated by a multi-dimensional Reflected Brownian Motion (RBM) with drift. Standard performance measures such as system occupancy, response times and server utilization are derived from the stationary probability distributions of the RBM. However, classical queueing theory usually does not consider networks carrying real-time traffic and their performance, nor real-time scheduling disciplines such as EDF. The reader should consult, for example, [4, 6, 12]. These papers (especially Peterson [12] for the feed-forward network case) develop Reflected Brownian Network models for multi-dimensional workloads at each node. Note, the RBM model does not depend upon the interarrival and service distributions, only their first two moments. Hence this theory is much more general and widely applicable than Jackson networks. It should be noted that all of the above mentioned papers focus on the FIFO scheduling discipline. For the purpose of real-time scheduling, we need to approximate the workload processes of the network when all the node are scheduled under the EDF algorithm. It is shown in [15] that under certain conditions, the limiting workload processes assuming an EDF scheduling discipline at each node are the same as those under FIFO networks, hence they can be well approximated by a multi-dimensional RBM. These conditions are:

1. all the packets from all the applications have the same end-to-end deadline distribution \(G\).

2. all the packets arriving to the system from outside the network arrive to a single node (node 0 in Figure 1).

3. none of the flows merge at any node inside the network.

The network defined in Figure 3 satisfied conditions (2) and (3). Hence, if all the deadlines are generated by the same distribution, \((W_0, W_1, \ldots, W_N)\) is a multi-dimensional RBM as well. One can use [4] to determine numerically the equilibrium distribution of the Brownian Network model. In some cases, these models are of product form, in analogy with certain Jackson networks in queueing network theory. [6] develops conditions under which the equilibrium RBM network model has product form. The presence of a product form equilibrium distribution dramatically simplifies the calculation of packet lateness, since we need only consider the marginal distributions at each node. In equilibrium, those marginal distributions will be exponential with a parameter depending upon the moments of the interarrival and service distributions of the streams using the particular node in question.

The third step in the analysis is to combine the real-time system performance (calculated from the leadtime profiles determined in the first step) with the equilibrium distribution. By integrating against the equilibrium distribution, we can determine the marginal performance, for example the percentage of packets that will be late as a function of the deadline structure of the streams. This integration step can be performed in closed form in many cases provided the equilibrium distribution is of product form. If product form does not hold, then one must perform the integration numerically. We next present cases in which the equilibrium distribution of the multi-dimensional reflected Brownian motion is of product form.
3.1 Product Form Solution

In general, the equilibrium probability distribution of a multi-dimensional RBM is the solution of a certain partial differential equation which must be solved numerically [4]. However, if the first two moments of the interarrival time and service time distributions satisfy certain conditions, the probability density of the RBM is of product form [6]. For our network shown in Figure 3, in order to satisfy the product form criteria, we need

$$\sum_{i=(j-1)k}^{jk} \lambda_i a_i^2 m_{iA} m_{iB}$$

(3.1)

for \( j = 1, 2, \cdots, N \). Under the above condition, for \( \mathbf{x} = (x_0, x_1, \cdots, x_N), x_i \geq 0, \) for all \( i \), the steady distribution \( p(\mathbf{x}) \) of \( Z \) is

$$p(\mathbf{x}) = \prod_{j=0}^{N} \gamma_j \exp(-\gamma_j x_j)$$

(3.2)

where

$$\gamma_0 = \frac{2(1-\rho_0)}{\sum_{i=1}^{NK} \lambda_i (a_i^2 + b_i^2_A) m_{iA}^2}$$

$$\gamma_j = \frac{2(1-\rho_j)}{\sum_{i=(j-1)k}^{jk} \lambda_i (a_i^2 + b_i^2_B) m_{iB}^2}$$

(3.3)

for \( j = 1, 2, \cdots, N \). The product form probability distribution provides us with an easy way to derive the relevant real-time performance measures. This is shown in next section.

4 End-to-End Delay Analysis

In this section, we develop analytic formulas to predict the end-to-end delay of individual packets. This delay will depend upon the deadlines associated with arriving packets and the scheduling policy used at each node. The real-time feed-forward queueing network theory developed by [15] shows (in both the EDF and the FIFO cases) that there is a deterministic relationship between the workload in the system and the lead-time of a departing packet. The deterministic relationship, described below, depends upon the deadline distribution of the arriving packets. Packets departing with a negative lead-time are late. Thus, we need only determine the workloads under which the lead-time of a departing packet is negative. This, in turn, gives us a mapping between workload and packet lateness. The probability that a packet is late is the probability that the workload is large enough so that departing packets are late. Hence, the proportion of late packets for each session can be approximated by the probability that the workload exceeds a certain threshold. Hereafter, for simplicity and in order to make use of the product form workload distribution, we assume that all the packets have the same end-to-end deadline distribution \( G(\eta) \).

Let

$$\tilde{W}_j = \frac{W_j}{\rho_j} + \frac{W_0}{\rho_0}$$

for \( j = 1, \cdots, N \).

In our network model, \( \tilde{W}_j \) can be thought as the total time that a packet spends in the system. An important quantity for determining lateness is the tail probability of \( \tilde{W}_j \). By convolving the probability distributions given by (3.2), we can compute the probability distribution of \( \tilde{W}_j \). This results in

$$P(\tilde{W}_j > x) = \frac{\gamma_0 \rho_0 \exp(-\gamma_j \rho_j x) - \gamma_j \rho_j \exp(-\gamma_0 \rho_0 x)}{(\gamma_0 \rho_0 - \gamma_j \rho_j)}$$

(4.1)

In the following, we provide formulas for the proportion of late packets for each session when all the nodes are scheduled under either EDF or FIFO respectively.

4.1 EDF Scheduling

We first consider the case in which all of the nodes in the network are scheduled using EDF. Suppose the lead-time of the packet exiting node \( j \) is \( F_j \), also called the “frontier”. [15] showed that when the traffic intensities converge to one, \( F_j \) satisfies

$$\tilde{W}_j = \int_{\eta_j}^{\infty} (1-G(\eta)) d\eta$$

(4.2)

for \( j = 1, \cdots, N \). We can use the above relationship between the workload, \( \tilde{W}_j \), and the frontier, \( F_j \), to determine the miss rate (proportion of late packets) of each session, when the end-to-end deadline distribution is given. In the following, we provide a closed form formula for the session miss rate in two cases, constant and uniform deadline distributions.

Case 1: Constant Deadline \( D \)

If all of the packets of all of the sessions have a common constant deadline, \( D \), then FIFO scheduling is equivalent to EDF scheduling. Given \( \tilde{W}_j, F_j \) can be determined by substituting \( G(\eta) = I(\eta \geq D) \) in (4.2), where
For $j = 1, \ldots, N$. Packets exiting from node $j$ miss the deadline if and only if $F_j < 0$. This condition is equivalent to $\bar{W}_j > D$. Hence, by (4.2), the session miss rate is

$$\frac{\gamma_0 \rho_0 \exp(-\gamma_j \rho_j D) - \gamma_j \rho_j \exp(-\gamma_0 \rho_0 D)}{\gamma_0 \rho_0 - \gamma_j \rho_j}.$$  (4.4)

**Case 2: Uniform Deadline $(A, B)$**

Let the end-to-end deadline of each packet be $U$, where $U$ is a random variable with a uniform $(A, B)$ distribution. The lead-time of a packet departing the system from node $j$ is $U - \bar{W}_j$. Consequently, a packet exiting from node $j$ is late if and only if $\bar{W}_j > U$. The lateness probabilities can be computed by the convolution of deadline distribution and the workload distribution governed by (3.2). In this case, the session miss rate is

$$\frac{\gamma_j \rho_j}{\gamma_0 \rho_0} \left( e^{-\gamma_0 \rho_0 A} - e^{-\gamma_0 \rho_0 B} \right) - \frac{\gamma_0 \rho_0}{\gamma_j \rho_j} \left( e^{-\gamma_j \rho_j A} - e^{-\gamma_j \rho_j B} \right).$$

**5 Simulation Results**

To validate the prediction formulas presented in Section 4, we performed extensive simulation experiments on the distributed network described in Section 2 with two nodes and two sessions ($K = 1, N = 2$). We assumed Poisson external arrivals for session 1 and 2 with rates $\lambda_1$ and $\lambda_2$ respectively. All the service time distributions are Gamma($\alpha, \beta$) with the corresponding means such that the traffic intensities at all the nodes are 0.95, which reflects the heavy traffic conditions. In particular, $m_{1A} = m_{2A} = 1$. The gamma distribution is chosen because it can produce service time with different values of coefficients of variation, as opposed to exponential service times whose c.v. is always 1.

The results of this study are presented in Tables 1, 2 and 3. We consider two cases: the first one with arrival rates $\lambda_1 = .6$ and $\lambda_2 = .35$, whereas the second case has more symmetric arrival rates of $\lambda_1 = .5$ and $\lambda_2 = .45$. There are three sections in the table. The first part presents a Jackson network case, i.e., all the service times are exponential. Note that for Jackson networks, the product form workload distribution is an exact solution. The second and third parts of the table show the general networks where the service times follow Gamma distributions. As the means are fixed, the variance of the Gamma distributions are chosen so that the coefficients of variation $(b_1, b_2)$ satisfy the product form criteria (3.2).

The first column gives the coefficients of variation for the two sessions at their local nodes, and these values have no restriction in order to obtain product form solutions. The last column shows the prediction which made using the real-time queuing network methodology developed earlier in this paper. The simulation results report the result of 50 independent runs of 100 million total task completions (for a total of 5 billion task completions). For each task completion, we
note whether or not it meets its deadline and compute the resulting miss rate over the simulation run. By combining the 50 runs, we can produce a sample average and a 95% confidence interval, which are shown in the third column of the table. Tables 2 and 3 are very similar in nature. Both assume that the deadlines of all arriving tasks are random with a Uniform(300,500) distribution. Table 2 presents the results for EDF scheduling at all nodes. Table 3 presents the results for FIFO scheduling at all nodes. Of course, Table 1 has constant deadlines, so the two scheduling methods are equivalent.

The results are very striking and illustrate the excellent accuracy that can be achieved using the real-time queuing network methodology. In many cases, the prediction is within the confidence interval. In other cases, the prediction is not inside, but the order of magnitude for the miss rate is correct. These results illustrate the great potential for this stochastic real-time scheduling methodology.

Table 1: End-to-end delay prediction with constant deadline = 400.

<table>
<thead>
<tr>
<th>(λ₁, λ₂)</th>
<th>K</th>
<th>Simulation</th>
<th>Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.5, 0.45)</td>
<td>1</td>
<td>7.12e-6 (3.79e-6, 1.04e-5)</td>
<td>8.89e-6</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>9.80e-4 (9.82e-4, 1.03e-3)</td>
<td>9.99e-4</td>
</tr>
<tr>
<td>(0.45, 0.45)</td>
<td>1</td>
<td>5.76e-5 (4.66e-5, 6.80e-5)</td>
<td>5.66e-5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.48e-4 (1.39e-4, 1.67e-4)</td>
<td>1.46e-4</td>
</tr>
<tr>
<td>(0.6, 0.35)</td>
<td>1</td>
<td>4.38e-5 (2.08e-5, 2.64e-5)</td>
<td>4.38e-5</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1.03e-3 (0.55e-3, 1.98e-3)</td>
<td>1.00e-3</td>
</tr>
</tbody>
</table>

Table 2: End-to-end delay prediction with Uniform (300,500) deadline, EDF scheduling.

<table>
<thead>
<tr>
<th>(λ₁, λ₂)</th>
<th>(b₁², b₂²)</th>
<th>K</th>
<th>Simulation</th>
<th>Theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.6, 0.35)</td>
<td>(1.5, 1.5)</td>
<td>1</td>
<td>4.38e-5 (2.08e-5, 2.64e-5)</td>
<td>4.38e-5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2</td>
<td>1.03e-3 (0.55e-3, 1.98e-3)</td>
<td>1.00e-3</td>
</tr>
<tr>
<td>(0.45, 0.45)</td>
<td>(1.8, 1.8)</td>
<td>1</td>
<td>2.18e-3 (2.06e-3, 2.54e-3)</td>
<td>2.64e-3</td>
</tr>
</tbody>
</table>

6 Admission Controls

For real-time systems, an admission control policy (or acceptance test) must be implemented to decide whether a new session can be admitted or whether such an admission will result in some sessions missing their deadlines. Each new session arriving to the system must specify its traffic characteristics and its required QoS. A new session is admitted only if all sessions (the newcomer and those that are already in the system) are able to maintain their required QoS. Traditionally, admission control policies are designed based on deterministic real-time scheduling theory to provide absolute guarantees for hard real-time tasks. Such policies often lead to an under-utilization of the system. As we will show in this section, control policies based on real-time queueing theory can achieve much higher system utilization. Of course, a stochastic formulation will result in the possibility of packet lateness, rather than the absolute guarantees offered by hard real-time scheduling theory. Thus, there is a tradeoff between high resource utilization and some packet lateness. For sessions that require a statistical guarantee, the corresponding QoS would be in terms
Table 3: End-to-end delay prediction with Uniform (300,500) deadline, FIFO scheduling.

<table>
<thead>
<tr>
<th></th>
<th>Jackson network</th>
<th>General network</th>
<th>General network</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(λ₁, λ₂)</td>
<td>(λ₁, λ₂ = .35)</td>
<td>(λ₁ = .5, λ₂ = .45)</td>
</tr>
<tr>
<td></td>
<td>K</td>
<td>Simulation</td>
<td>Theory</td>
</tr>
<tr>
<td>(6,35)</td>
<td>1</td>
<td>2.9e-5</td>
<td>3.3e-5</td>
</tr>
<tr>
<td>(5,45)</td>
<td>1</td>
<td>1.67e-3</td>
<td>1.67e-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(2.9e-3,3.3e-4)</td>
<td></td>
</tr>
<tr>
<td>(2,2)</td>
<td>1</td>
<td>7.9e-4</td>
<td>7.9e-4</td>
</tr>
<tr>
<td>(1,1)</td>
<td>1</td>
<td>4.28e-5</td>
<td>3.29e-5</td>
</tr>
<tr>
<td>(5,5)</td>
<td>1</td>
<td>1.85e-3</td>
<td>1.67e-3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(3.71e-5,4.85e-5)</td>
<td></td>
</tr>
</tbody>
</table>

of bounds on the session miss rates. Hence, the miss rate formulas developed in the previous section can be used to design admission policies that provide statistical delay bounds. In this section, we introduce session admission control policies that use the real-time queuing network approach.

We consider a VoD network shown in Figure 3, with N local nodes and K sessions at each local node. The traffic pattern generated by each session is assumed to be identical with the following characteristics:

- interarrival times are uniformly distributed on the interval \((p - \delta, p - \delta)\), where \(0 \leq \delta \leq p\).
- execution times are distributed uniformly on \((m_A - \Delta_1, m_A + \Delta_1)\) at node 0 and are distributed uniformly on \((m_B - \Delta_2, m_B + \Delta_2)\) at the local nodes, where \(0 \leq \Delta_1 \leq m_A\) and \(0 \leq \Delta_2 \leq m_B\).
- each job has a constant end-to-end deadline \(D\).
- jobs are scheduled at each node according to EDF.

The above traffic model represents a video stream that generates frames periodically with period \(p\) and an amount of jitter \(\delta\), and the processing/transmission time of each frame is variable on a bounded interval. Note that when \(\delta = \Delta_1 = \Delta_2 = 0\), it is a periodic task model.

To support an absolute guarantee with no packet lateness, we use hard real-time scheduling approaches [10, 11] for the admission control. The traffic of each session is then assumed to follow the end-to-end periodic task model with constant interarrival time (period) \(p_i = p - \delta\) and constant execution times \(e_i = m_A + \Delta_1\) and \(e_i = m_B + \Delta_2\) at the central node and local nodes respectively. The jobs are scheduled locally at the central node and local nodes according to the EDF algorithm, with local deadlines \(D_A\) and \(D_B\) respectively assigned by the proportional deadline algorithm [13]. That is \(D_A = D - \frac{\alpha_i}{\sum \alpha_i}\) and \(D_B = \frac{\alpha_i}{\sum \alpha_i}\). To determine whether the given system of \(NK\) independent periodic traffic streams meet all the end-to-end deadlines, we check whether the inequalities \(\sum_{i=1}^{NK} \frac{\gamma_i}{\min(D_A, p_i)} \leq 1\) and \(\sum_{i=1}^{K} \frac{\gamma_i}{\min(D_B, p_i)} \leq 1\) are satisfied. Given the degree of the jitter and the variation of execution times (i.e. \(\delta, \Delta_1, \Delta_2\)), one can determine the maximum number of session admitted while providing the absolute QoS guarantees.

We next turn to application admission control policy based on real-time queueing theory. This policy will determine, for a given packet miss rate, the number of sessions that can be serviced by this network. The formulas needed for this were developed in the previous section. For each session \(i = 1, 2, \cdots, NK\),

1. The traffic pattern for Session \(i\) is specified by \((\lambda_i, a_i^2, m_i, b_i^2, \beta_i^2, \alpha_i^4)\). In addition, its deadline miss rate bound \(\alpha_i\) must also be specified.
2. We next compute \(\gamma_i, \rho_i\) according to (3.3).
3. We compute the miss rate for each session according to (4.4).
4. If miss rate of session \(i\) is less than \(\alpha_i\) for all \(i\), the system can provide the required QoS for the given number of channels.

Again, given the traffic characteristics and the miss rate bound, one can determine the maximum number of sessions that the system can support and still provide the QoS guarantees, and hence the maximum average utilization at each node. It is important to notice that the above admission control policy does not require the specific interarrival time and computation time distribution, just the first two moments of these distribution. Such measurements are often easy to obtain either through experimental modeling or on-line monitoring.
To compare real-time queueing network theory with hard real-time scheduling, we compare the maximum average utilization when the admission controller uses the two different approaches. We consider an example with \( p = 18.9, m_A = 0.18, m_B = 1.8 \) being fixed. We look at the maximum number of sessions each local node can support while meeting the required absolute or statistical guarantees, across different values of \( \Delta_2 \), with other parameter being fixed. For the chosen \( \Delta_1 \), the value of \( \delta \) is picked so that the product form criteria (3.2) is satisfied. More specifically, we need \( \frac{\delta}{p} = \frac{\Delta_1}{m_A} \) for identical uniform interarrival time and service time distributions. In this case, we set \( \delta = 15.12 \) and \( \Delta_1 = 0.144 \). Figure 4 shows the maximum number of session supported by the local nodes for various \( \Delta_2 \) values, \( 0 \leq \Delta_2 \leq 2 \). The end-to-end deadline for each packet is 18.9, the period. The three curves represent the maximum value of \( K \) when providing three levels of required QoS (miss rate bounded at \( 0, 10^{-7} \) and \( 10^{-14} \) respectively). In Figure 5, we show the utilization gain at the local node when the system provides statistical guarantees rather than absolute guarantees. It is shown that even if the loss rate bound is as low as \( 10^{-14} \), the local nodes can achieve at least 50% utilization gain using RTQ vs using hard real-time scheduling. Figure 6 and 7 show the same results when the packet end-to-end deadlines are set to the maximum interarrival time, 34.02.

Figure 4: Maximum number of sessions at each local node, \( D = 18.9 \).

Figure 5: Utilization gain of statistical QoS, \( D = 18.9 \).

Figure 6: Maximum number of sessions at each local node, \( D = 34.02 \).

rate of a particular session is recorded. The dash curves represent the average and the 95% confidence intervals based on 10 independent runs, with a 10 million total task completions in each run. The real-time queueing network prediction is shown as a solid line. When the end-to-end deadline level is over 400ms, the prediction is within the 95% confidence interval.

The results presented in this section are very surprising. The simulation results clearly illustrate how inefficient resource utilization can be when one adopts the hard real-time scheduling methodology, but the tasks exhibit substantial variability in their interarrival times and computation requirement. Using real-time queueing network theory, it is possible to analytically determine the number of applications a system can support while keeping the deadline miss rate to a very low rate.
many sessions can be supported so that the real-time requirements (deadlines) of tasks are met up to a specified tolerance level. Simulation results attest to the excellent accuracy achieved by the real-time queueing network methodology.

The paper presents new results concerning the Brownian network approximation with EDF scheduling policy, in particular conditions are given under which the equilibrium distribution is of product form. Although the session miss rate formula presented in the paper requires traffic parameters to satisfy the product form criteria, the restriction is unnecessary provided one can compute the RBM distribution numerically.

While the model given in the paper are only presented for relatively simple networks, the methodology is quite general, and can be applied to much more complicated situations. For example, it is straightforward to generalize the results to the network with multiple levels of local nodes. Most importantly, the methodology is insensitive to the specific distributions that describe the interarrival pattern of tasks or their execution times.

In summary, this paper presents clear evidence that real-time queueing network methods provide an approach for controlling real-time systems in a way that the ability of the system to meet the timing requirements of the tasks using the system can be accurately predicted.

References


