Queueing Model and Optimization of Packet Dropping in Real-Time Wireless Sensor Networks

Marc Aoun*, Antonios Argyriou†,
*Philips Research, Eindhoven, 5656AE, The Netherlands
†Department of Computer and Communication Engineering, University of Thessaly, Volos, 38221, Greece

Abstract—**We consider the problem of modeling the transmission of real-time data from a single node of a wireless sensor network to the next hop or access point. Generated packets are placed in a single position buffer and are transmitted over the wireless medium to the next hop, or they are discarded and replaced when a new data packet is generated from the sensor. To increase the percentage of packets delivered on-time to the next hop, we introduce a thresholding policy that is supported by an analytical model and it is responsible for deciding whether to transmit a data packet or drop it and transmit the next one. Our model considers the behavior at single sensor where data are generated with a Poisson process and have a fixed deadline associated to them while the impact of the other sensors is modeled through their effect on the MAC service time. However, we also evaluate the performance of the packet dropping scheme in a multi-hop sensor chain. The proposed policy leads to a higher percentage of on-time delivered packets in both scenarios.**

Index Terms—**Real-time communication, wireless sensor networks, packet dropping, queuing model.**

I. Introduction

In wireless sensor networks (WSNs) real-time data generation and transmission dominate the majority of application scenarios. One of the main characteristics of sensor data that is generated in real-time is that their value may become obsolete when a new one is generated. In this paper, we investigate the communication of real-time messages by developing a simple queueing theory model and we use it for further optimization. We model the transmitter of a sensor node as a single-server queue. The generated data packets have to be transmitted within a certain deadline. The queueing system is assumed to have an overwrite-buffer with space for one packet; when a new packet bearing more recent data arrives to the queueing system, the buffered packet, if any, is discarded and the new packet is stored in the buffer. Including the buffer space at the transmitter, the system represents an M/M/1/2 queue with deadlines. The important contribution of this work to the aforementioned model is that it introduces a thresholding mechanism. According to this mechanism, a buffered data packet is dropped (removed from the buffer) if the remaining time until its deadline is shorter than a certain threshold θ. The intuition behind this policy is that a packet is not transmitted if there is low probability of being received before its deadline expires, saving thus transmission time for a newly arriving packet and in addition wireless bandwidth. By dropping the currently stored packet, and waiting on a new packet arrival instead, the percentage of on-time packets is increased because a newly arriving packet has a higher probability of meeting its deadline. Furthermore, with this proactive approach a packet that would be characterized lost in the next hop, if it was late, will not be transmitted over the wireless medium and consume bandwidth.

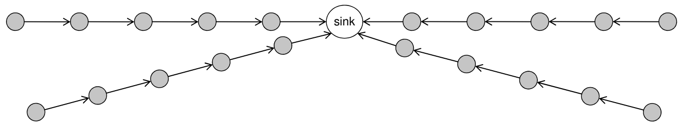


Fig. 1: Multiple wireless sensors generate real-time data that are delivered to a sink. With multi-hop communication the wireless nodes closer to the sink may become the bottleneck since they have to forward higher portion of the traffic.

II. Related Works

To the best of our knowledge, the problem of modeling the real-time packet transmission in WSNs has not been adequately addressed. Nevertheless, the second problem we are interested in this paper, i.e. the phenomenon of overloading the core of a WSN that operates under multiple sources and one sink has been reported in several works [1]. Fig. 1 depicts a simple wireless sensor network where problems like the above may occur depending on the locally generated traffic load at each sensor. Optimization efforts in this type of wireless sensors networks have focused on reducing the traffic load and also on designing medium access control (MAC) protocols that prioritize the nodes that are closer to the sink [2], [3]. Nevertheless these mechanisms attempt to address the problem from the perspective of the better utilization of the network without looking at the sources themselves and how to prevent unnecessary data from being injected into the network.

However, for optimizing the injection of real-time traffic into the network, there is a need for analytical performance models that characterize the impact of all possible actions taken by a node. The works in this area are limited and focus primarily on the queuing model itself without considering its applicability in a network setting. Barrer was one of the first to analyze an $M/M/1$ queue with deterministic deadlines [4], [5]. This analysis was extended to systems with state-dependent arrival and service rates, and more general deadlines by Brandt and Brandt [6], [7] and by Movaghar [8]. These

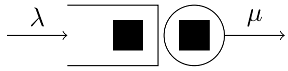


Fig. 2: A schematic representation of the model for a sensor node that adopts proactive packet dropping.

studies all deal with non-preemptive systems with deadlines to service beginnings, but in [9], Movaghar considers deadlines to service endings with preemption. Movaghar and Kargahi [10], [11] have devised an approximation for an $M/M/1$ queue with the earliest-deadline-first (EDF) discipline, which is known to stochastically maximise the fraction of packets served before their deadline (see, e.g., [12]). Lehoczky [13] analyzed an $M/M/1$ queue with deadlines to service endings and the EDF-policy in heavy traffic. He argued that, since the deadlines of all stored packets have to be taken into account, this queue gives rise to a Markov process on a statespace of infinite dimension. Lehoczky shows that the Markov process collapses to a tractable one-dimensional process in heavy traffic. Lehoczky later used these results to analyze control policies in [14], and extended his analysis to Jackson networks in [15]. Doytchinov et al. [16] extended this analysis to a $GI/G/1$ queue with deadlines, and Kruk et al. [17] to networks of such queues.

III. Performance Analysis

A. Sensor System Model

We now provide further details for the performance analysis of the proposed system model. Packets are generated at each sensor according to a Poisson process with parameter λ. The packets have exponentially distributed service times with parameter μ that correspond to the behavior of the MAC protocol and the impact of channel contention from other nodes. A packet transmission is considered successful if it is *completed* within d time units after the arrival of the packet to the queue, i.e., if a packet arrives at time t, its transmission must finish before its global deadline $t + d$, with d being a fixed relative deadline. Packet transmission at the physical layer (PHY) is non-preemptive, meaning that if a packet starts being transmitted it will finish regardless of whether another packet arrives. We define the packet delivery ratio (PDR) of the system as the fraction of packets that are transmitted successfully, *i.e. within the deadline requirement*.

B. PDR Calculation

In this section, we compute the PDR of the system in the presence of packet dropping denoted by $\gamma(\theta)$. The parameter θ denotes the packet dropping threshold given in seconds, and it will be discussed in more detail later. We denote by L the number of packets present in the system (including the transmitter) immediately before a packet arrival. A packet arriving to the system will find the latter in one of the following three states:

- *Empty (L = 0)*: The transmitter is available, and there is no packet occupying the buffer spot.
- *Busy Transmitter (L = 1)*: The transmitter is transmitting a packet but the buffer is empty.
- *Full, (L = 2)*: The transmitter is busy and the buffer also contains a packet. In that case, an arriving packet overwrites the buffered one.

The PDR $\gamma(\theta)$ is by definition equal to the probability that the transmission of an arbitrary packet is successful. To compute this probability, we condition on whether a packet arrives at an empty system or not. Denoting by B the service time that the arriving packet would experience, We have:

$$\gamma(\theta) = \mathbb{P}(L=0)\mathbb{P}(B \leq d) + (1 - \mathbb{P}(L=0)) \int_0^{d-\theta} \mu e^{-\mu t} \mathbb{P}(B \leq d-t) e^{-\lambda t} dt \quad (1)$$

The rationale behind Eq. (1) is as follows: If an arbitrary packet P arrives at an *empty* system, it is transmitted successfully only if the required transmission time B is less than d. This explains the term $\mathbb{P}(L = 0)\mathbb{P}(B \leq d)$. If P arrives at a *non-empty* system ($L = 1$ or $L = 2$), we condition on the length of the remaining (residual) transmission time t of the packet that is currently being transmitted. The residual service time is exponentially distributed, so its density is $\mu e^{-\mu t}$. Moreover, given that the residual service time is t, the service of P is successful if the following three conditions are all met: First, the time until the deadline is larger than θ when the residual service ends, i.e., $d - t \geq \theta$. This is taken into account in the integration region of Eq. (1). Second, the service time is less than $t - d$, which explains the factor $\mathbb{P}(B \leq t - d)$. Third, there were no other arrivals during t time units, which explains the term $e^{-\lambda t}$.

C. Probability of an Empty System

Next, we calculate $\mathbb{P}(L = 0)$, i.e. the probability that an arbitrary packet arrives at an empty system. For this purpose we define the *cycle time C* which is the time between two consecutive time instants at which the system becomes empty and it is depicted in Fig. 3. The cycle time is separated into two periods: A first *idle period* during which the system is empty, and a *busy period* during which the transmitter is occupied until it becomes empty again. The busy period spreads throughout the time that the system is not in a state $L = 0$. It consists of a collection of consecutive states characterized by $L = 1$ or $L = 2$. We shall refer to the case where $L = 2$ during a busy period as *clearance period* (CP), which is the amount of time the buffer is continuously occupied. A clearance period ends when the buffer is cleared, i.e. when either a dispatch to the transmitter occurs, or when the stored packet is denied service and removed from the system following the thresholding procedure.

The idle time lasts until a packet arrives, so the mean idle time is $1/\lambda$. If the busy period is denoted by BP, we have:

$$\mathbb{E}[C] = \frac{1}{\lambda} + \mathbb{E}[BP]. \quad (2)$$

From the PASTA-property [18], we have that the probability that a packet arrives at an empty system, $P(L = 0)$,

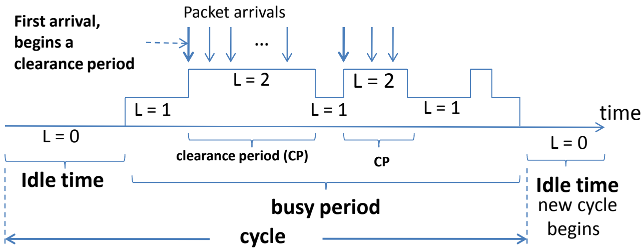


Fig. 3: A cycle consisting of idle time and busy period.

it is equal to the probability that the system is empty at an arbitrary time. To derive this probability we divide the mean idle time by the mean cycle time as follows:

$$\mathbb{P}(L=0) = \frac{1/\lambda}{\mathbb{E}[C]} = \frac{1/\lambda}{1/\lambda + \mathbb{E}[BP]}. \quad (3)$$

Now we derive an expression for the busy period. The busy period starts when a packet arrives to an empty system. After this, the expected time until the first event (an arrival or service completion) is $1/(\lambda+\mu)$. Furthermore, with probability $\mu/(\lambda+\mu)$, the first event is a service completion, and the busy period ends after $1/\mu$ time units. With probability $\lambda/(\lambda+\mu)$, the first event is an arrival. In this case, the arriving packet is stored in the buffer and the buffer has to be cleared, which takes in expectation $\mathbb{E}[T]$ time units. Once the buffer has been cleared, the expected time before the system becomes empty is again equal to $\mathbb{E}[BP]$ due to the memorylessness of service times. From the above we have that the mean busy period is

$$\mathbb{E}[BP] = \frac{1}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu}\left(\mathbb{E}[T] + \mathbb{E}[BP]\right), \quad (4)$$

where $\mathbb{E}[T]$ is the mean buffer clearance period. We will derive $\mathbb{E}[T]$ next. With algebraic manipulations (4) yields

$$\mathbb{E}[BP] = \frac{1}{\mu} + \frac{\lambda}{\mu}\mathbb{E}[T]. \quad (5)$$

Now, it remains to determine $\mathbb{E}[T]$, the expected length of the buffer clearance period. Without loss of generality, we assume that the buffer clearance period starts at time 0. We condition on t the time at which the next event (arrival or service completion) occurs. Because the time until the next event is exponentially distributed with parameter $\lambda+\mu$, we obtain:

$$\begin{aligned}\mathbb{E}[T] &= \int_{d-\theta}^{\infty} (\lambda+\mu)e^{-(\lambda+\mu)t}(d-\theta)dt \\ &+ \int_{0}^{d-\theta} \frac{\mu}{\lambda+\mu}(\lambda+\mu)e^{-(\lambda+\mu)t}tdt \\ &+ \int_{0}^{d-\theta} \frac{\lambda}{\lambda+\mu}(\lambda+\mu)e^{-(\lambda+\mu)t}(t+\mathbb{E}[T])dt.\end{aligned}$$

The three integrals (from left to right) cover the following three possibilities: First, if there are no events before time $d-\theta$, the buffer is cleared at time $d-\theta$ because the deadline of the buffered packet becomes smaller than θ. Second, if the first event occurs at time $t < d-\theta$, and the first event is a service completion, the buffer is cleared at time t because the buffered packet enters service. Third, if the first event occurs at time $t < d-\theta$ and is a packet arrival, the arriving packet overwrites the buffered packet, and a new buffer clearance period begins.

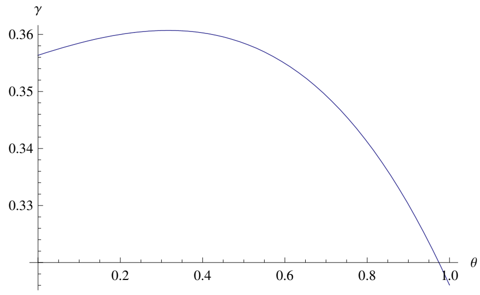


Fig. 4: The PDR as a function of θ, with $\mu=\lambda=d=1$.

After rewriting and evaluating the integrals (the rightmost two using partial integration), we obtain:

$$\mathbb{E}[T] = \frac{1-e^{-(\lambda+\mu)(d-\theta)}}{\mu+\lambda e^{-(\lambda+\mu)(d-\theta)}}. \quad (6)$$

Combining Equations (3), (5), and (6) we have that

$$\mathbb{P}(L=0) = 1 - \frac{\lambda^2+\lambda\mu}{\mu^2+\lambda\mu e^{-(\lambda+\mu)(d-\theta)}+\lambda^2+\lambda\mu} \quad (7)$$

Finally, by substituting (7) into (1) and evaluating the integrals we have that the PDR $\gamma(\theta)$ is given by

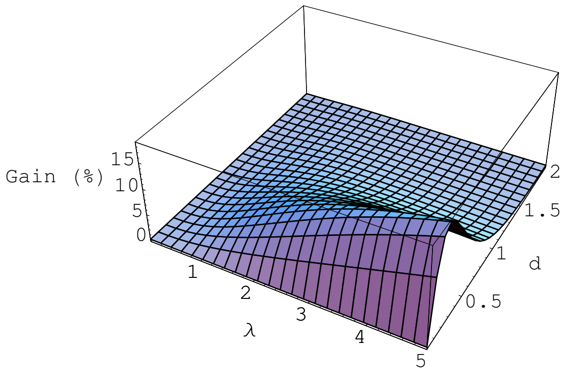
$$\begin{aligned}\gamma(\theta) &= \mathbb{P}(L=0)(1-e^{-\mu d}) + \\ &(1-\mathbb{P}(L=0))\Big[\frac{\mu}{\lambda+\mu}\left(1-e^{-(\lambda+\mu)(d-\theta)}\right) \\ &- \frac{\mu e^{-\mu d}}{\lambda}\left(1-e^{-\lambda(d-\theta)}\right)\Big]. \quad (8)\end{aligned}$$

Note that if $d\to\infty$, deadlines become irrelevant and two special cases occur for specific values of θ. If $\theta=0$, packets in the buffer are always transmitted, regardless of the time until their deadline. The PDR of the system is thus equal to that in an $M/M/1/2$ queue (see e.g., [19, Section 5.7]. If $\theta=d\to\infty$, packets in the buffer are never transmitted. Only packets arriving to an empty system are. In this case, the PDR is equal to that in an $M/M/1/1$ queue. By substituting d and θ, these values follow from Equation (8).

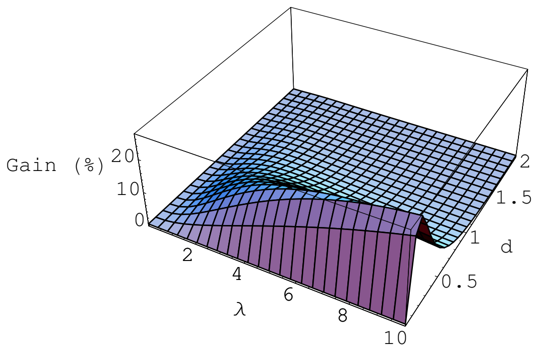
IV. Numerical Results for a Single Sensor

In this section, we study the PDR numerically. In Fig. 4, we present $\gamma(\theta)$ for various values of θ, with $\mu=\lambda=d=1$. We clearly see that the PDR of the system is indeed increased by the threshold θ, as long as θ is chosen appropriately.

Having established that an appropriately chosen threshold increases PDR, we study *how much* the PDR can be increased by this threshold. Note that $\gamma(0)$ is the PDR of the system



(a) $0 \leq \lambda \leq 5$.



(b) $0 \leq \lambda \leq 10$.

Fig. 5: Maximal PDR improvement.

without thresholding. We define θ^* as the threshold that maximizes the relative increase in PDR, i.e., we define

$$\theta^* = \arg\max \left\{ \frac{\gamma(\theta) - \gamma(0)}{\gamma(0)} : 0 \leq \theta \leq d \right\}.$$

We define the maximal PDR improvement as the maximal relative increase in PDR, i.e., as $(\gamma(\theta^*) - \gamma(0))/\gamma(0)$.

In the sequel, we set $\mu = 1$. We can make this assumption without loss of generality; all parameters are relative to each other, so for any set of parameters, we can scale time in such a way that $\mu = 1$ without changing the PDR of the system. In Fig. 5(a) we display the maximal PDR improvement for d between 0 and 2, and λ between 0 and 5. Fig 5(a) implies that, in this parameter region, we can obtain an increase of up to 15% in PDR by setting the threshold to its optimal value. Furthermore, we see that the relative increase in PDR is maximal if $d \approx 0.3$, but even for larger values of d there can be a PDR enhancement. In addition to this, the maximal PDR improvement grows as λ grows, so the PDR policy is especially beneficial if the system is overloaded.

Likewise, in Fig. 5(b) we show the maximal PDR improvement for λ up to 10. In this region, the increase in PDR can be as much as 30%. Furthermore, the value of d with the largest relative increase in PDR, as well as the range of d for which gain is achieved become smaller. This is conveyed by Fig. 6, which plots the gain for three increasing values of λ, equal to 3, 4 and 6. The narrower range of d where gain is achieved is explained by a shorter sojourn time of buffered packets that later enter service, due to more frequent buffer overwriting for an increasing arrival rate. The shorter sojourn time reflects itself in a larger lead-time when the buffered packet is assessed, by the packet dropping mechanism, for dispatch to the transmitter. This in turn presents itself in a decrease of the upper value of d for which packet dropping still results in transmission denials for some packets, and as such, still results in PDR gain.

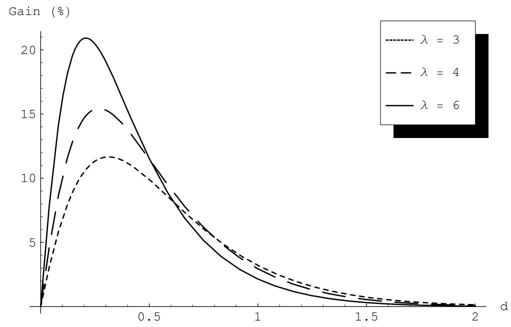


Fig. 6: PDR gain for λ values equal to 3, 4 and 6.

We also present values of the improvement for specific values of λ and d in Table I for a single sensor. We clearly see that, for this parameter region, the maximal PDR increase grows as λ becomes larger. Furthermore, given a fixed deadline, the maximal improvement increases up to a certain value of λ, and decreases beyond that value.

V. Numerical Results for a Sensor Chain

According to our setup, each node in the sensor chain generates locally a load λ and also forwards the successfully received data packets from the previous node. Here, we assume that this total load follows again an exponential distribution. This assumption is necessary since if adopt a generalized distribution for the arrival rate at a node, this will make the problem intractable. The closed form analysis analysis was only possible for a single queue as we have seen in the previous sections.

n=1		d				
		0.1	0.2	0.3	0.4	0.5
λ	0.5	0.23	0.37	0.45	0.48	0.48
	1	1.04	1.71	2.07	2.21	2.18
	2	1.036	5.90	6.91	7.05	6.64
	4	10.07	14.69	15.30	13.78	11.49
	8	21.36	24.20	19.61	14.31	10.08

TABLE I: The maximal PDR improvement (%) for specific values of λ and d.

n=5 nodes		d 0.1	0.2	0.3	0.4	0.5
	0.5	1.23	1.37	2.01	2.10	2.07
	1	2.43	2.84	4.2	5.41	5.66
λ	2	4.52	10.20	13.91	15.39	14.75
	4	21.47	30.02	32.67	34.78	32.02
	8	49.16	48.67	44.61	42.64	39.98

n=10 nodes		d 0.1	0.2	0.3	0.4	0.5
	0.5	3.42	3.3	2.99	2.80	2.48
	1	8.49	7.98	6.34	6.01	5.99
λ	2	10.21	11.27	17.11	16.87	15.31
	4	50.78	39.21	34.46	32.66	29.49
	8	75.89	65.41	62.34	60.08	58.08

TABLE II: The maximal PDR improvement (%) for specific values of λ and d and for sensor chains of n=5 and n=10 nodes.

We now present values of the improvement for specific values of λ and d in Table II for a chain of 5 and 10 sensors. Regarding the results, we see that as the number of nodes in the chain is increased, the load that must be forwarded naturally increases. Therefore, contrary to the case of low λ and $n = 1$ (seen in Table I) where the PDR improvement of the proposed scheme is not that important, in the case of $n = 5$ it is. The reason is that each node has to forward an increased traffic load and not only the low λ that is generated locally. As the number of nodes becomes even higher, and in our case n becomes 10, the PDR improvement becomes even higher. Recall that the results are relative to the case that does not apply proactive packet dropping. The same trend follows for higher d where the significant performance benefits are preserved. The reason for the reduced importance of d is due to the fact that the λ dominates the system behavior.

VI. Conclusions

In this paper, we developed an analytical model that captures the impact of proactive packet dropping on the transmission of real-time packets from a single sensor node. We showed that with the use of the analytical model, an optimal dropping policy can be identified, leading to PDR improvement of up to 30% for a single hop. When this model is applied in the case of a multi-hop chain network of sensors, the PDR improvement is more significant. This is predominantly due the ability of our model to predict the delayed arrival of a packet and subsequently drop it before transmission. We plan to extend our model to a system that exhibits different service time distributions and larger buffer size. Nevertheless, as it seems from our current analysis these extensions are not trivial.

References

[1] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and L. Mo, "Does wireless sensor network scale? a measurement study on greenorbs," in *IEEE INFOCOM*, 2011.

[2] S. Olariu and I. Stojmenovic, "Design guidelines for maximizing lifetime and avoiding energy holes in sensor networks with uniform distribution and uniform reporting," in *IEEE INFOCOM*, 2006.

[3] X. Wu, G. Chen, and S. K. Das, "Avoiding energy holes in wireless sensor networks with nonuniform node distribution," *IEEE TPDS*, 2008.

[4] D. Barrer, "Queueing with impatient customers and indifferent clerks," *Operations Research*, vol. 5, no. 5, pp. 644–649, 1957.

[5] ——, "Queueing with impatient customers and ordered service," *Operations Research*, vol. 5, no. 5, pp. 650–656, 1957.

[6] A. Brandt and M. Brandt, "On the m(n)/m(n)/s queue with impatient calls," *Performance Evaluation*, vol. 35, no. 1-2, pp. 1–18, 1999.

[7] ——, "Asymptotic results and a Markovian approximation for the m(n)/m(n)/s+gi system," *Queueing Systems*, vol. 41, no. 1-2, pp. 73–94, 2002.

[8] A. Movaghar, "On queueing with customer impatience until the beginning of service," *Queueing Systems*, vol. 29, no. 4, pp. 337–350, 1998.

[9] ——, "On queueing with customer impatience until the end of service," *Stochastic Models*, vol. 22, no. 1, pp. 149–173, 2006.

[10] M. Kargahi and A. Movaghar, "Non-preemptive earliest-deadline-first scheduling policy: A performance study," in *IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems*, Georgia, ATL, USA, 2005, p. 201210.

[11] ——, "A method for performance analysis of earliest-deadline-first scheduling policy," *Journal of Supercomputing*, vol. 37, no. 2, pp. 197–222, 2006.

[12] S. Panwar, D. Towsley, and J. Wolf, "Optimal scheduling policies for a class of queues with customer deadlines to the beginning of service," *Journal of the ACM*, vol. 35, no. 4, pp. 832–844, 1988.

[13] J. Lehoczky, "Real-time queueing theory," in *IEEE Real-Time Systems Symposium*, Washington, DC, USA, 1996, p. 186195.

[14] ——, "Using real-time queueing theory to control lateness in real-time systems," *Performance Evaluation Review*, vol. 25, no. 1, pp. 158–168, 1997.

[15] ——, "Real-time queueing network theory," in *IEEE Real-time Systems Symposium*, San Francisco, CA, USA, 1997, p. 5867.

[16] B. Doytchinov, J. Lehoczky, and S. Shreve, "Real-time queues in heavy traffic with earliest-deadline-first queue discipline," *The Annals of Applied Probability*, vol. 11, no. 2, pp. 332–378, 2001.

[17] L. Kruk, J. Lehoczky, S. Shreve, and S.-N. Yeung, "Earliest-deadline-first service in heavy-traffic acyclic networks," *The Annals of Applied Probability*, vol. 14, no. 3, pp. 1306–1352, 2004.

[18] R. Wolff, "Poisson arrivals see time averages," *Operations Research*, vol. 30, no. 2, pp. 223–231, 1982.

[19] ——, *Introduction to Queueing Theory*. Prentice-Hall, Inc., 1989.