A REGENERATION-BASED APPROACH FOR RESOURCE ALLOCATION IN COOPERATIVE DISTRIBUTED SYSTEMS

Sagar Dhakal, Jorge E. Pezoa, and Majeed M. Hayat

Electrical and Computer Engineering and Center for High Technology Materials
University of New Mexico
Albuquerque, NM 87131-0001 USA; E-Mail: {dhakal, jpezoa, hayat}@ece.unm.edu

ABSTRACT

A regeneration-based approach is undertaken to derive an expression for the expected value of total service time for an arbitrary customer distribution when scheduling is performed in a distributed queuing system. The service times and the transfer delays for customers in the shared communication medium are both considered random. The theory is applied to a distributed wireless-sensor network and the interplay between the total service time and the energy consumption of each sensor is investigated.

Index Terms— regeneration theory, distributed systems

1 Introduction

A multidimensional queuing framework for distributed systems can be utilized to analyze the performance of grid computing systems, distributed sensor networks as well as telecommunication networks [1, 2]. In a multidimensional queuing model, a collection of distributed servers communicate over a shared medium, where customers arrive randomly in time and space. Scheduling of customers, or load balancing, is performed among the servers in order to reduce the average queuing time per customer. Due to limitations in communication resources, however, transfer of customers from more-busy servers to less-busy servers is always accompanied by random delays. In addition, the amount of energy spent on scheduling as well as the associated delays both depend upon the size of scheduling, the specifics of the communication medium, and the locations of source and destination servers involved in the scheduling process. Therefore, care must be exercised so that no excessive amount of time and energy is unduly wasted in scheduling while the collective processing power of the distributed system is utilized maximally. There is therefore a fundamental tradeoff between savings in queuing time per customer, resulting from utilizing the processing power of a distributed system cooperatively, and the combined delay and energy overhead resulting from the very collaborative nature of the servers.

In this paper we exploit the concept of regeneration in stochastic processes [3,4] to analyze the dynamics of the queues that evolve after customer-scheduling is performed in a multidimensional queuing model. To this end, we derive an expression for the expected value of total service time for an initial number of customers. Our theoretical approach can be useful in solving complex queuing problems that arise in several important areas such as distributed computing and sensor networks, among other problems. For illustration, we apply the theory to develop an optimal scheduling policy for energy-limited distributed sensor networks.

2 Queuing model

Consider a system of n distributed servers connected over a network of arbitrary topology. Let $m_k \geq 0$ be the initial number of customers of the kth server at time $t = 0$, where the service time of each customer is random. In order to divide the total customers of the system among all servers, a synchronized scheduling action is performed at time $t = 0$ so that each server, the kth server, say, transfers an amount $L_{jk} \geq 0$ customers to the jth server ($j \neq k$). Naturally, these customer exchanges that occur over the shared communication network take random transfer times.

Definition of network state: Let $g_k \in \{0, 1, 2, 3, ...\}$ represent the number of different groups of customers that are simultaneously in transit to the kth server at time t. We assign a vector $\mathbf{c}_k$ of size $g_k + 1$ such that each component of $\mathbf{c}_k$ (except the first component) represents the number of customers within a particular group that is in transit, while the first component of $\mathbf{c}_k$ is always set to g_k. We now define the *network state* as the concatenated vector $\mathbf{C} \triangleq (\mathbf{c}_1, \ldots, \mathbf{c}_n)$. For example, in a 3-server system, $\mathbf{C} = ([2\ 3\ 2], [1\ 10], [0])$ at time t corresponds to the network state for which two different groups of customers (3 customers in the first group and 2 customer in the second group) are being transferred to the first server ($\mathbf{c}_1 = [2\ 3\ 2]$), one group of 10 customers is being transferred to the second server ($\mathbf{c}_2 = [1\ 10]$), while there is no transfer being made to the third server ($\mathbf{c}_3 = [0]$).

Let $T_{r_1,\ldots,r_n}(\mathbf{C})$ denote the total service time for all customers in the system if the network state at $t = 0$ (just after

This work was supported by the National Science Foundation under Award ANI-0312611.

scheduling is performed) is as specified by $\mathbf{C}$, while r_k customers ($k = 1, \dots, n$) remain at the kth server at $t = 0$. Our objective is to calculate $\mathsf{E}[T_{r_1,\dots,r_n}(\mathbf{C})]$.

Assumption A1 (Exponential distribution of delays): The following random variables are assumed to be exponentially distributed: (i) W_{ki}: the service time for the ith customer of the kth server (with rate λ_{d_k}); (ii) Z_{kj}: the arrival time of the jth group of customers sent to the kth server (with rate λ_{ki}).

Assumption A2 (Independence of delays): W_{ki} and Z_{ki} are assumed to be mutually independent for all i, k, j.

Convention C1 (Degenerate cases): The following time delays are set to ∞ almost surely (a.s.): (i) service time at any server with no customers, and (ii) the customer-arrival time when there is no group of customers in transit.

Assumptions A1 and A2 are well approximated according to our earlier empirical data obtained from load-balancing experiments (over a wireless LAN) for distributed-computing applications [5]. In addition, those experiments also revealed that the mean arrival time for the ith group of q customers sent to the kth server is $\lambda_{ki}^{-1} = pq$, where $p > 0$ is an experimentally calculated channel constant (in seconds per task).

2.1 Regeneration time

The key idea of stochastic regeneration in our approach is to introduce a special random variable, called the *regeneration time*, τ, defined as the minimum of the following two random variables: the time to the first customer service by any server, or the time to the first arrival of a group of customers at any server. More precisely, $\tau \triangleq \min\big(\min_k(W_{k1}), \min_{k,i}(Z_{ki})\big)$. Note that in light of Assumptions A1, A2, and Convention C1, it is straightforward to see that τ is an exponentially distributed random variable. The key property of the regeneration time τ is that the occurrence of the regeneration event $\{\tau = s\}$ gives birth to a new queuing system at $t = s$ whose random times satisfy Assumptions A1 and A2 while having its own initial system condition. The new initial system condition can be a new initial customer distribution if the regeneration event is a service to a customer, or a new customer distribution and a new $\mathbf{C}$ if the regeneration event is an arrival of a group of customers.

2.2 Regenerative equations

Our main result for $\mathsf{E}[T_{r_1,\dots,r_n}(\mathbf{C})]$ is in the form of a regenerative equation as given in Theorem 1. Note that $\mathsf{E}[T_{0,0,\dots,0}([0], [0], \dots, [0])] = 0$ since $T_{0,0,\dots,0}([0], [0], \dots, [0]) = 0$ a. s., since there are no customers to be serviced in the system.

Theorem 1: For $n \in \mathbb{N}, r_k \in \mathbb{Z}^+$, $g_k \geq 0$ and $c_{ki} > 0$, the following recursion holds:

$$\begin{aligned}
&\mathsf{E}\big[T_{r_1,\dots,r_n}\big([g_1\ c_{11}\dots c_{1g_1}], \dots, [g_n\ c_{n1}\dots c_{ng_2}]\big)\big] = \\
&\frac{1}{\lambda} + \sum_{k=1}^{n} \frac{\lambda_{d_k}}{\lambda} \mathsf{E}\Bigg[T_{r_1-\delta_{1,k},\dots,r_n-\delta_{n,k}}\Bigg([g_1\ c_{11}\dots c_{1g_1}], \dots, \\
&[g_n\ c_{n1}\dots c_{ng_2}]\Bigg)\Bigg] \\
&+ \sum_{k=1}^{n}\sum_{i=1}^{g_k} \frac{\lambda_{ki}}{\lambda} \mathsf{E}\Bigg[T_{r_1+\delta_{1,k}c_{ki},\dots,r_n+\delta_{n,k}c_{ki}}\Bigg([g_1\ c_{11}\dots \\
&c_{1g_1}], \dots, [g_k - 1\ c_{k1}\dots c_{k(i-1)}\ c_{k(i+1)}\dots c_{kg_k}], \\
&\dots, [g_n\ c_{n1}\dots c_{ng_2}]\Bigg)\Bigg],
\end{aligned} \tag{1}$$

where $\lambda = \sum_{k=1}^{n}(\lambda_{d_k} + \sum_{i=1}^{g_k}\lambda_{ki})$, $\delta_{j,k}$ is the Kronecker delta and $r_k - 1$ is set to 0 when $r_k = 0$.

Proof:
By exploiting the smoothing property of the conditional expectation, we can write:

$$\begin{aligned}
&\mathsf{E}[T_{r_1,\dots,r_n}(\mathbf{C})] = \int_0^\infty \Bigg(\sum_{k=1}^{n} \mathsf{E}[T_{r_1,\dots,r_n}(\mathbf{C})|\tau = s, \tau = W_{k1}] \\
&\mathsf{P}\{\tau = W_{k1}|\tau = s\} + \sum_{k=1}^{n}\sum_{i=1}^{g_k} \mathsf{E}[T_{r_1,\dots,r_n}(\mathbf{C})|\tau = s, \tau = Z_{ki}] \\
&\mathsf{P}\{\tau = Z_{ki}|\tau = s\}\Bigg) f_\tau(s)ds,
\end{aligned} \tag{2}$$

where, $f_\tau(t) = \lambda e^{-\lambda t}u(t)$ is the pdf of τ with $\lambda = \sum_{k=1}^{n}(\lambda_{d_k} + \sum_{i=1}^{g_k}\lambda_{ki})$. Let $T'_{r_1,\dots,r_n}(\mathbf{C})$ denote the total service time of the new queuing system that has emerged at time τ with the network state $\mathbf{C}$ and $r_k \geq 0$ customers ($k = 1, \dots, n$) in the queue of the kth server. It is straightforward to show that $\mathsf{E}\big[T_{r_1,\dots,r_n}(\mathbf{C})|\tau = s, \tau = W_{k1}\big] = \mathsf{E}\big[\tau + T'_{r_1,\dots,r_k-1,\dots r_n}(\mathbf{C})$ $|\tau = s, \tau = W_{k1}\big] = s + \mathsf{E}\big[T'_{r_1,\dots,r_k-1,\dots r_n}(\mathbf{C})|\tau = s, \tau = W_{k1}\big]$. Similarly, we can write $\mathsf{E}\big[T_{r_1,\dots,r_n}(\mathbf{C})|\tau = s, \tau = Z_{ki}\big] = \mathsf{E}\big[\tau + T'_{r_1,\dots,r_k+c_{ki},\dots r_n}(\mathbf{C}')|\tau = s, \tau = Z_{ki}\big] = s + \mathsf{E}\big[T'_{r_1,\dots,r_k+c_{ki},\dots r_n}(\mathbf{C}')|\tau = s, \tau = Z_{ki}\big]$, where $\mathbf{C}'$ is same as $\mathbf{C}$ but without c_{ki} since c_{ki} customers have now joined the queue of the kth server.

Finally, it turns out that conditional on the occurrence of $\{\tau = s, \tau = W_{ki}\}$ or $\{\tau = s, \tau = Z_{ki}\}$, the random times of the new queuing system at time $t = s$ satisfy Assumptions A1 and A2. Therefore, we can shift the time origin from $t = 0$ to $t = s$, without changing the statistics of the random delays, which leads to the following results: $\mathsf{E}\big[T'_{r_1,\dots,r_k-1,\dots r_n}(\mathbf{C})|\tau = s, \tau = W_{k1}\big] = \mathsf{E}\big[T_{r_1,\dots,r_k-1,\dots r_n}$ $(\mathbf{C})\big]$ and $\mathsf{E}\big[T'_{r_1,\dots,r_k+c_{ki},\dots r_n}(\mathbf{C}')|\tau = s, \tau = Z_{ki}\big]$ $= \mathsf{E}\big[T_{r_1,\dots,r_k+c_{ki},\dots r_n}(\mathbf{C}')\big]$. Now, by using these results in (2) and noting that $\mathsf{P}\{\tau = W_{k1}|\tau = s\} = \frac{\lambda_{d_k}}{\lambda}$ and $\mathsf{P}\{\tau = Z_{k1}|\tau = s\} = \frac{\lambda_{ki}}{\lambda}$, we complete the proof of Theorem 1. □

3 Application to Wireless Sensor Networks

Wireless sensor networks typically consist of small-battery powered processors deployed over a region. These sensors communicate with each other over radio links. In some situations, few sensors might be overloaded by collecting data at a high rate while others remain idle. This could lead to a situation where the network looses some sensing coverage as the overloaded sensors become rapidly depleted of battery power. In addition, due to computational limitations of a sensor, it may not be possible for the sensor to process its data in a timely fashion. Thus, by allowing the sensors to process the raw data cooperatively we may not only extend the lifetime of some batteries but also enhance the computing efficiency of the sensor network. However, as we stated earlier, transferring data between sensors requires energy, and these transfers will be accompanied by random delays.

3.1 Scheduling policies for two cooperating sensors

Consider a two-node sensor network. At time $t = 0$, the first sensor (overloaded) transfers L_{21} number of tasks to the second sensor (idle) using the following policy:

$$L_{21} = \lfloor K m_1 \rfloor, \tag{3}$$

where $K \in [0,1]$ is the parameter defining the scheduling policy and $\lfloor x \rfloor$ is the greatest integer less than or equal to x. Suppose that for any particular value of K, the average energy consumed for processing, transferring and receiving tasks by the jth sensor is ε_j. The *minimum–service-time (MST) policy* is the scheduling policy that minimizes the expected value of the total service time. More precisely, the MST policy is defined by the optimal K that minimizes $\mathsf{E}\big[T_{r_1,r_2}([0],[1\ L_{21}])\big]$ for $r_1 = m_1 - L_{21}$ and $r_2 = m_2$. The *fair-energy (FE) policy* is the scheduling policy that ensures equal energy consumption for each sensor. More precisely, the FE-policy is defined by the fair K that achieves $\varepsilon_{\text{fair}} \triangleq \varepsilon_1 = \varepsilon_2$.

Example 1: Suppose that at time $t = 0$, the first node has sensed data equivalent to 100 tasks and the second node is idle, i.e., m_1 = 100 and m_2 = 0. (A task is defined as the amount of data required by a preprocessing algorithm in order to compute one value of a desired quantity and processing of one task is one execution of the preprocessing algorithm.) Suppose that the service rates (in task per second) are $\lambda_{d_1} = 1$ and $\lambda_{d_2} = 0.5$. Further, let the channel constant be $p = 0.2$ s per task. Moreover, by adopting the radio-energy model for an actual sensor [6], we set the energy dissipation rate for each sensor to be 1 mJ per task for transmission, and 0.5 mJ per task for reception. Finally, the energy dissipation rates for task processing at the first and second sensors are 4 mJ per task and 2 mJ per task, respectively.

In Fig. 1, we see that the expected value of the total service time, calculated according to the Theorem, attains its minimum value of 74 s for $K = 0.28$. If we assume an infinite transfer rate, the fair amount of tasks to be transferred to the second node would only depend upon the processing rates of the sensors, and it is given by $\frac{\lambda_{d_2}}{\lambda_{d_1}+\lambda_{d_2}} \times 100$, which is approximately equal to 33 tasks. Instead, with a transfer-rate of 5 tasks per second, the MST policy (corresponding to $K = 0.28$) transfers only 28 tasks to the second node in order to avoid excess delay in the channel. However, it should also be noted that for any particular value of K, and according to our assumptions on energy consumption, $\varepsilon_1 = L_{21} + 4r_1$ and $\varepsilon_2 = 0.5L_{21} + 2(r_2 + L_{21})$. Next, we can observe from Fig. 2 that at $K = 0.28$, the energy consumption per sensor becomes unfair as the first sensor consumes about 4.5 times the energy consumed by the second sensor. However, the FE policy, corresponding to $K = 0.73$, results in $\varepsilon_{\text{fair}} = 182$ mJ of energy consumption at each sensor; nonetheless, at $K = 0.73$, the expected value of the total service time is about 161 s.

In order to jointly investigate the effect of K on the interplay between the total service time and the energy consumption of each sensor, we define the following normalized quantities that, respectively, measure the policy's deviation from the points of minimum service time and fair-energy consumption:

$$\sigma_{T_{21}} = \frac{\mathsf{E}\big[T_{r_1,r_2}(\mathbf{C})\big] - T_{\text{min}}}{T_{\text{max}}}, \quad \sigma_{\epsilon_{21}} = \frac{\sqrt{\sum_{l=1,2}(\varepsilon_l - \varepsilon_{\text{fair}})^2}}{2\varepsilon_{\text{max}}},$$

where $T_{\text{min}} \triangleq \inf\Big\{\mathsf{E}\big[T_{r_1,r_2}(\mathbf{C})\big], K \in [0,1]\Big\}$, $T_{\text{max}} \triangleq \sup\Big\{\mathsf{E}\big[T_{r_1,r_2}(\mathbf{C})\big], K \in [0,1]\Big\}$ and $\varepsilon_{\text{max}} = \max_K\big(\max(\varepsilon_1,\varepsilon_2)\big)$. In order to achieve a fair tradeoff between the deviation in service time and deviation in energy consumption per sensor, we introduce a scheduling policy called the *fair-tradeoff (FT)* policy that is given by K yielding $\sigma_{T_{21}} = \sigma_{\epsilon_{21}}$. In the case of Example 1, the FT policy is given by $K = 0.5$ for which $\varepsilon_1 = 250$ mJ, $\varepsilon_2 = 125$ mJ and the expected value of the total service time is approximately 110 s.

3.2 Extension to n cooperating sensors

In an n-node system, the excess tasks at the jth sensor at time $t = 0$, L_j^{ex}, is calculated as: $L_j^{ex} = m_j - \frac{\lambda_{d_j}}{\sum_{k=1}^n \lambda_{d_k}} \sum_{l=1}^n m_l$, [5]. Following [7], let $\mathcal{V} \triangleq \{j : L_j^{ex} > 0\}$ be the collection of candidate sender sensors. Next, for any $j \in \mathcal{V}$, the excess load of the jth sensor is partitioned among sensors having tasks smaller than the average tasks per sensor. Let $\mathcal{U} \triangleq \{i : L_i^{ex} < 0\}$ be the collection of such candidate recipient sensors. The partition p_i $(i = 1,\ldots,n)$ is now defined as

$$p_i = \begin{cases} L_i^{ex} / \sum_{l\in\mathcal{U}} L_l^{ex}, & i \in \mathcal{U} \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Finally, in order to compensate for the transfer delay, the fair share of the ith sensor, $p_i L_j^{ex}$, is reduced to L_{ij} using the following scheduling policy determined by K_{ij}:

$$L_{ij} = \lfloor K_{ij} p_i L_j^{ex} \rfloor, \ K_{ij} \in [0,1]. \tag{5}$$

Calculation of K_{ij} : At first, we arbitrarily choose one recipient sensor, say the ith sensor, that belongs to $\mathcal{U}$. Then, we set $K_{kj} = 1$ for all $k \in \mathcal{U}\setminus\{i\}$, while the objective is to calculate

K_{ij}. Now, in the context of the ith and the jth sensors, the scheduling policy is defined by K_{ij} and the total service time is $\mathsf{E}\big[T_{r_j,r_i}([0],[1\ L_{ij}])\big]$ for $r_j = m_j - \sum_{k\in\mathcal{U}} \lfloor K_{kj} p_k L_j^{ex} \rfloor$, $r_i = m_i$. We can now efficiently calculate K_{ij} based on either the MST, the FE or the FT policies.

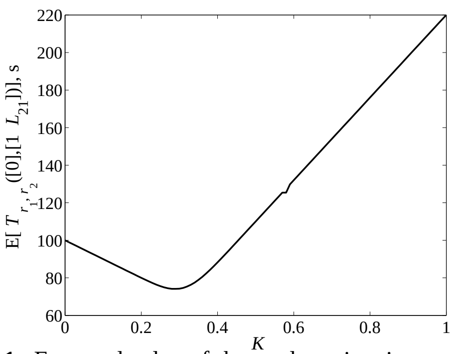

Figure 1: Expected value of the total service time under different scheduling policies.

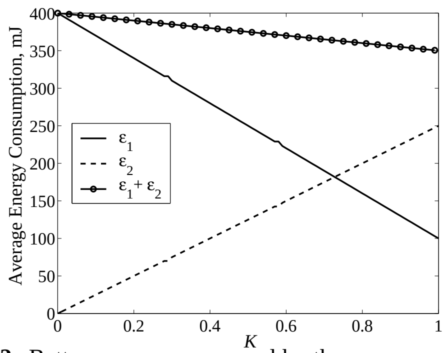

Figure 2: Battery-energy consumed by the sensors under different scheduling policies.

In the next step, we arbitrarily choose another recipient node, say the lth node, that belongs to $\mathcal{U}\setminus\{i\}$ and set $K_{kj} = 1$ for all $k \in \mathcal{U} \setminus \{i,l\}$, while we utilize K_{ij} obtained from the previous step. The expected value of total service time is $\mathsf{E}\big[T_{r_j,r_l}([0],[1\ L_{lj}])\big]$ where $r_j = m_j - \sum_{k\in\mathcal{U}} \lfloor K_{kj} p_k L_j^{ex} \rfloor$ and $r_l = m_l$. Next, we calculate K_{lj} based on either MST, FE or FT policies. These steps are repeated until we calculate K_{ij} for all $i \in \mathcal{U}$ and for all $j \in \mathcal{V}$.

Example 2: Consider a five-node sensor network for which λ_{d_1}, λ_{d_2}, λ_{d_3}, λ_{d_4}, and λ_{d_5} (in units of tasks per second) are 0.25, 0.5, 0.75, 1 and 1.25, respectively, while the energy dissipation rates for processing (in units of mJ per task) at the first to fifth sensors are 0.5, 1, 1.5, 2 and 2.5, respectively. The load-transfer rate as well as the energy dissipation rates for transmission and reception are set as given in Example 1. The initial task-distribution is: $m_1 = 500, m_2 = \ldots = m_5 = 0$. The average service time (AST) and the average energy consumption (AEC) per sensor under the MST, FE and FT policies are listed in Table 1. The MST policy gives a small AST compared to FE and FT policies, but the aggregate AEC of five sensors is much larger. Similarly, the AST under the FE policy is much larger compared to the other two. In summary, the FT policy offers a well-balanced performance in terms of reducing the AST and the AEC together.

Table 1: The AST and the AEC per sensor under the three policies defined.

MST policy		FE policy		FT policy	
AST (s)	AEC (mJ)	AST (s)	AEC (mJ)	AST (s)	AEC (mJ)
188.5	$\varepsilon_1 = 481$ $\varepsilon_2 = 99$ $\varepsilon_3 = 200$ $\varepsilon_4 = 330$ $\varepsilon_5 = 459$	1091.6	$\varepsilon_1 = 363$ $\varepsilon_2 = 75$ $\varepsilon_3 = 100$ $\varepsilon_4 = 145$ $\varepsilon_5 = 207$	616.0	$\varepsilon_1 = 423$ $\varepsilon_2 = 86$ $\varepsilon_3 = 154$ $\varepsilon_4 = 243$ $\varepsilon_5 = 345$

4 Conclusions

We have considered a multidimensional queuing model for a cooperative system of servers where each server executes a synchronized scheduling action. Our model specifically captures the effects of random service time for each customer and random transfer delays in the communication network. To track customer transit at any given time, we have introduced the concept of network state vector. Based on the assumption that all the random delays follow exponential distributions, we have invoked a novel regeneration argument to characterize the expected value of the total service time for a given initial customer distribution.

We have applied the theory to design novel scheduling policies for a distributed sensor network application. The interplay between the expected value of the total service time and the energy consumption of each sensor is highlighted. To this end, we have considered three different scheduling policies: (1) the minimum-service-time policy that minimizes the expected value of the total service time, (2) the fair-energy policy that ensures a fair consumption of energy by each sensor, and (3) the fair-tradeoff policy that jointly considers the service time and the deviation in energy consumption per sensor. Our preliminary results for a two-node and five-node sensor networks indicate that the fair-tradeoff policy achieves a well-balanced performance as compared to the service-optimal and fair-energy policies in terms of reducing the average service time and the average energy consumption.

5 References

[1] M.J. Neely, E. Modiano, and C.E. Rohrs, “Dynamic power allocation and routing for time varying wireless networks,” in *Proc. IEEE INFOCOM*, San Francisco, April 2003.

[2] D.Y. Burman and D.R. Smith, “A light traffic theorem for multiserver queues,” *Mathematics of Operational Research*, 8:15-25, 1983.

[3] C. Knessly and C. Tiery, “Two tandem queues with general renewal input I: Diffusion approximation and integral representation,” *SIAM J. Appl. Math.*, vol. 59, pp. 1917–1959, 1999.

[4] F. Bacelli and P. Bremaud, *Elements of queuing theory: Palm-martingale calculus and stochastic recurrence.* New York: Springer-Verlag, 1994.

[5] S. Dhakal, M.M. Hayat, J.E. Pezoa, C. Yang, and D.A. Bader, “Dynamic load balancing in distributed systems in the presence of delays: A regeneration-theory approach,” *IEEE Trans. Parallel and Distributed Systems*, 2006, in press.

[6] J. Hill, R. Szewczyk, A.Woo, S. Hollar, D. Culler, and K. Pister, “System architecture directions for networked sensors,” in *Proc. Int. IX Conf. ASPLOS*, Cambridge, MA, 2000.

[7] S. Dhakal, M.M. Hayat, and J.E. Pezoa, “Reliability in distributed queuing systems in the presence of random delays,” submitted to *IEEE Trans. Inf. Theory*, 2006.