Real-Time Status: How Often Should One Update?

Sanjit Kaul*‡, Roy Yates† and Marco Gruteser†

*IIIT Delhi, India, E-mail: skkaul@iiitd.ac.in

†WINLAB, Rutgers University, NJ, USA, E-mail:{ryates,gruteser}@winlab.rutgers.edu

Abstract—**Increasingly ubiquitous communication networks and connectivity via portable devices have engendered a host of applications in which sources, for example people and environmental sensors, send updates of their status to interested recipients. These applications desire status updates at the recipients to be *as timely as possible*; however, this is typically constrained by limited network resources. In this paper, we employ a time-average age metric for the performance evaluation of status update systems. We derive general methods for calculating the age metric that can be applied to a broad class of service systems. We apply these methods to queue-theoretic system abstractions consisting of a source, a service facility and monitors, with the model of the service facility (physical constraints) a given. The queue discipline of *first-come-first-served (FCFS)* is explored. We show the existence of an optimal rate at which a source must generate its information to keep its status as timely as possible at all its monitors. This rate differs from those that maximize utilization (throughput) or minimize status packet delivery delay. While our abstractions are simpler than their real-world counterparts, the insights obtained, we believe, are a useful starting point in understanding and designing systems that support real time status updates.**

I. INTRODUCTION

The *information age* [1] has witnessed huge improvements in computing, access and storage of information. More recently, fueled by ubiquitous connectivity and advancements in portable devices, *real-time status* updates have become increasingly popular. These range from news and weather reports and updates by individuals on Twitter about what is keeping them busy, to updates by environmental sensors [2].

Real-time status updates can enable a variety of applications. Temperature and humidity updates from a forest can help better predict and control forest fires, energy utilization information can help make a smart-home energy efficient, knowledge of the velocity, acceleration of a car, see Figure 1, can assist drivers in an intelligent transportation system to make safe maneuvers [3].

In the above examples, the goals are to ensure that the agency that monitors fires stays current about conditions in the forest and drivers stay current about status of vehicles in their vicinity, respectively. These examples share a common description: a source generates time-stamped status update messages that are transmitted through a communication system to a monitor. The goal of real-time status updating is to ensure that the *status* of interest, is *as timely as possible* at each monitor. When the monitor's most recently received update at time t is time-stamped $u(t)$, the *status update age*, which we will refer to as simply the *age*, is $t - u(t)$. The monitor's

---

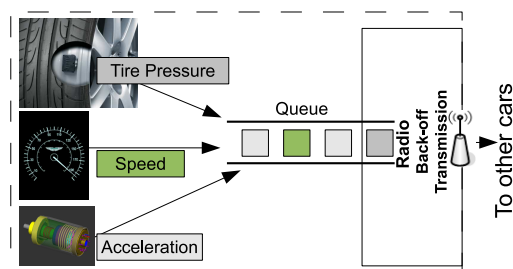‡Sanjit was with WINLAB, Rutgers University, when he did this work.

Fig. 1: Sensors in a vehicle generate their status updates, which the in-vehicle radio delivers to other vehicles in vicinity. The status updates from a sensor are queued as is, or aggregated with other sensors' updates, to be served by the radio.

requirement of timely updating corresponds to small average status update age.

We will see that the goal of timely updating is neither the same as maximizing the utilization of the communication system, nor of ensuring that generated status updates are received with minimum delay. Utilization may be maximized by making the sensor send updates as fast as possible. However, this may lead to the monitor receiving delayed statuses because the status messages become backlogged in the communication system. In this case, delay suffered by the stream of status updates can be reduced by reducing the rate of updates. Alternatively, reducing the update rate can also lead to the monitor having unnecessarily *outdated* status information because of a lack of updates.

How often must a source generate status updates? Ideally, we would want a monitor to receive a status update, typically communicated in packet form, at the very instant it was generated at a source. If this were possible, a source would simply generate status updates as fast as possible. However, real world constraints dictate that the delivery of a status message requires a nonzero and typically random time in the system (network). In many systems, this time will also depend on the previously queued packets in the system.

Figure 1 shows an example of such a system. Status updates are generated by sensors in a vehicle. The status update packets are queued while they *wait* to be *serviced* by the car radio. The packet *currently* being serviced by the radio *waits* for medium access and transmission before it is received by other cars. Note that each sensor in the car may be a source or that the car may aggregate a collection of sensor measurements into a status update message that is transmitted as a single packet. To simplify the following discussion, we focus on this case and assume the car radio is carrying the packets of a single source.

Suppose the radio interface is a simple queue; packets passed to the radio are queued first-come-first-served (FCFS) for direct transmission to the monitor. The service time of each packet will depend on the wireless channel. This service time may or may not incorporate retransmissions due to channel errors and backoff due to the activity of other wireless transmitters. While system models that incorporate these effects can be arbitrarily complex, we observe that even in simple settings in which service times are independent identically distributed (iid) random variables, for example an $M/M/1$ queue with iid exponential service times, the optimal updating rate is unknown.

In general, we will look at FCFS systems in which the real-time status *generation at the source* and the time spent by status packets in *service* before they are received by one or more monitors is described by known random processes.

We present an overview of related work in Section II. In Section III we derive the general expression of age for a FCFS queue. In Sections IV, V and VI we find server utilizations that minimize age for $M/M/1$, $M/D/1$ and $D/M/1$ systems, respectively. In Section VII we find the lower bound on achievable age for any service time distribution. We summarize our contributions and present possible extensions in Section VIII.

II. RELATED WORK

The requirement of ensuring freshness occurs in various fields, including that of networks, real time databases and warehousing. In [4] we look at minimizing the age of status updates sent by vehicles over a carrier-sense multiple access (CSMA) network. The minimum age can be approached using gradient descent. However, it is not known in general and is only seen to exist in simulations. In [5] we show that allowing nodes to piggyback other nodes' status updates can lead to a smaller age. In both [4] and [5] we use the phrase *system age of information* instead of age of status updates.

In [6] the authors want to maximize the freshness of data in warehouses to meet user demands. They estimate the queue length and delay at the staging area of a warehouse, which is where updates wait before they are committed to the warehouse database. Experiments lead them to conclude that they want to have small queue build ups. While their observation about queues is in line with ours, they do not consider optimal source (update) rates or systems in general.

Web caching reduces the latency in returning a web page to a client. However, unless refreshed often enough, a cache will return stale web pages. The rate of refreshing is limited by the finite time it takes for a cache to be updated after the page has been updated at the server. In [7] the authors propose an architecture that limits the "*degree of staleness*" of a cache. Our work, for fairly simple descriptions of the time it takes to update a cache, answers how often the cache must be refreshed such that its age is minimized.

In [8] the authors look at periodic transactions updating real time databases. Each transaction updates the database with data that is associated with a deadline relative to when it is generated. In their work, there is no assumed limit on available

processing power (service rate). The objective is to find the combination of update period and deadline that ensure that all transactions complete before their deadlines, ensuring the freshness of data and minimizing the CPU utilization.

Ad hoc networking protocols typically use a route cache to forward packets to their destinations. In [9] the authors propose a mechanism that avoids propagation of stale route information through the network. They do not want to broadcast new route information periodically, however, to avoid the associated overheads. Their method uses an epoch numbering system that helps a node in the network to reject older information. In [10] the authors consider the issue of frequency of hello messages in ad-hoc networks. The frequency must not be so large as to congest the network but also not too small that the nodes have stale information.

Finally, dissemination in sensor networks has been looked at under varied constraints. For example, in works like [11] and [12] the authors consider energy efficient dissemination of state in sensor networks. More frequent updates lead to greater energy consumption and smaller sensor lifetime. Our work suggests strategies that a sensor, when awake, can use to minimize age of its status updates.

III. FCFS STATUS UPDATE AGE

We derive the *average* status update age for a system that has a source updating a monitor through a *first-come-first-served* single server packet queue. We will use a graphical argument. Figure 2 shows a sample variation of age $\Delta(t)$, as a function of time t, at the monitor. Without loss of generality, assume that we begin observing at $t = 0$ when the queue is empty and that the age is $\Delta(0) = \Delta_0$. The first status update is generated at t_1, followed by updates at $t_2, t_3, \ldots, t_n$.

The age at the monitor increases linearly in time in the absence of any updates and is reset to a smaller value when an update is received. Update i, generated at time t_i, finishes service and is received by the monitor at time t_i'. At t_i', the age $\Delta(t_i')$ at the monitor is reset to the age $T_i = t_i' - t_i$ of the received status update. The age T_i is also the system time of the update packet i and is the sum of the time the packet waited in the queue and the time it spent in service. Thus the age function $\Delta(t)$ exhibits the sawtooth pattern shown in Figure 2.

The time average age of the status updates is the area under the sawtooth function in Figure 2 normalized by the time interval of observation. Over an interval $(0, \mathcal{T})$, the average age is

$$\Delta_{\mathcal{T}} = \frac{1}{\mathcal{T}} \int_0^{\mathcal{T}} \Delta(t) dt. \tag{1}$$

For simplicity of exposition, the length of the observation interval is chosen to be $\mathcal{T} = t_n'$, as depicted in Figure 2. We decompose the area defined by the integral in (1) into a sum of disjoint geometric parts. Starting from $t = 0$, the area can be seen as the concatenation of the polygon area $\tilde{Q}_1$, the trapezoids Q_i for $i \geq 2$ (Q_2 and Q_n are highlighted in the figure), and the triangular area of width T_n over the time
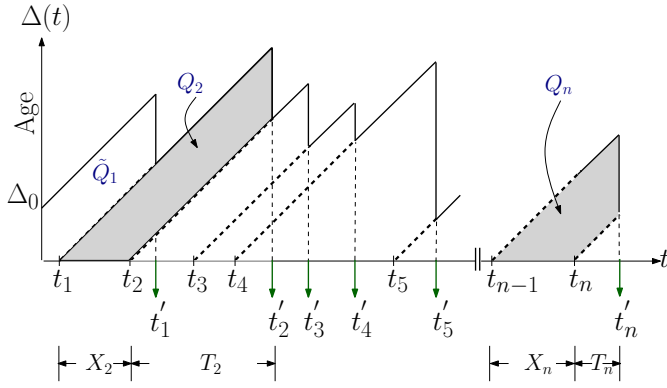
Fig. 2: Example change in age at a monitor under FCFS queuing.

interval (t_n, t'_n). With $N(\mathcal{T}) = \max\{n | t_n \leq \mathcal{T}\}$ denoting the number of arrivals by time $\mathcal{T}$, this decomposition yields

$$\Delta_{\mathcal{T}} = \frac{\tilde{Q}_1 + T_n^2/2 + \sum_{i=2}^{N(\mathcal{T})} Q_i}{\mathcal{T}}. \quad (2)$$

From Figure 2, we see that the area Q_i can be calculated as the difference between the area of the isosceles triangle whose base connects the points t_{i-1} and t'_i and the area of the isosceles triangle with base connecting the points t_i and t'_i. Defining

$$X_i = t_i - t_{i-1} \quad (3)$$

to be the elapsed time between the generation of updates $i-1$ and i, it follows that

$$Q_i = \frac{1}{2}(T_i + X_i)^2 - \frac{1}{2}T_i^2 = X_i T_i + X_i^2/2. \quad (4)$$

When the generation of updates can be represented as the arrivals of a stochastic process, X_i is the interarrival time of update i. Substituting (4) in (2), some rearrangement yields the time-average age

$$\Delta_{\mathcal{T}} = \frac{\tilde{Q}}{\mathcal{T}} + \frac{N(\mathcal{T}) - 1}{\mathcal{T}} \frac{1}{N(\mathcal{T}) - 1} \sum_{i=2}^{N(\mathcal{T})} \left[X_i T_i + \frac{X_i^2}{2} \right] \quad (5)$$

where, $\tilde{Q} = \tilde{Q}_1 + T_n^2/2$. We observe that the age contribution $\tilde{Q}$ represents a boundary effect that is finite with probability 1, so the first term in (5) will vanish as $\mathcal{T}$ grows. Let

$$\lambda = \lim_{\mathcal{T} \to \infty} \frac{N(\mathcal{T})}{\mathcal{T}} \quad (6)$$

be the steady state rate at which status update packets are generated. We assume that the limit exists and is finite. Furthermore, as $N(\mathcal{T}) \to \infty$, the remaining summation term in (5) is a sample average that will converge to its corresponding stochastic average. The average status update age can be obtained as

$$\Delta = \lim_{\mathcal{T} \to \infty} \Delta_{\mathcal{T}} = \lambda \left(E[XT] + E[X^2]/2 \right), \quad (7)$$

where $E[\cdot]$ is the expectation operator, and X and T are the random variables that correspond to the interarrival time and system time of an update packet, respectively.

We note that the average update age in (7) holds under weak assumptions on the ergodicity of the service system. Furthermore, (7) is a general result for a broad class of service systems in which the update packets are processed FCFS. For example, (7) would hold for a queue in which the status update stream shares the service facility with other packet streams. However, evaluation of the age Δ can be challenging. In particular, X is the random variable that describes the time between generation of an update packet and the one that preceded it while T is the system time of that same packet. The variables X and T are dependent. A large interarrival time X allows the queue to empty, yielding a small waiting time and typically a small system time T. That is, X and T tend to be negatively correlated and this complicates the evaluation of $E[TX]$.

Next we will find the age and the server utilization that minimizes it for standard queuing systems for FCFS queues. We start with the $M/M/1$ system.

IV. M/M/1 - FIRST COME FIRST SERVED

We consider the FCFS $M/M/1$ system with arrival rate λ and service rate μ. That is, update packets are generated and submitted to the system as a rate λ Poisson process and thus the status update interarrival times X_i are independent and identically distributed (iid) exponential random variables with $E[X] = 1/\lambda$. Further, service times are iid exponentials with average service time $1/\mu$. We will calculate the age for the system and then find the server *utilization* $\rho = \lambda/\mu$ that minimizes the average age Δ.

Consider the average age Δ in equation (7). For an exponential X, $E[X^2] = 2/\lambda^2$. We now will derive an expression for $E[TX]$. Consider status update i. Its system time is

$$T_i = W_i + S_i, \quad (8)$$

where W_i and S_i are the respective waiting time and service time of packet i. If packet $i-1$ has already been served when i is generated, $W_i = 0$. If i arrives when $i-1$ is either waiting for or receiving service, $W_i = T_{i-1} - X_i$. Combining these observations, we can write the waiting time of packet i as

$$W_i = (T_{i-1} - X_i)^+. \quad (9)$$

We note that the T_{i-1} depends on the arrivals and service times of packets prior to i and is independent of the ith interarrival time X_i. Furthermore, when the system reaches steady state the system times are stochastically identical, that is $T =^{st} T_i =^{st} T_{i-1}$. The probability density function of the system time T for the $M|M|1$ system is [13]

$$f_T(t) = \mu(1-\rho)e^{-\mu(1-\rho)t}, \qquad t \geq 0. \quad (10)$$

It follows that the conditional expected waiting time W_i given $X_i = x$ is

$$E[W_i | X_i = x] = E\left[(T_{i-1} - x)^+ | X_i = x\right] \quad (11)$$

$$= E\left[(T - x)^+\right] \quad (12)$$

$$= \int_x^\infty (t - x) f_T(t) \, dt = \frac{e^{-\mu(1-\rho)x}}{\mu(1-\rho)}. \quad (13)$$

Returning to the calculation of $E[TX] = E[T_iX_i]$, we note that the service time S_i is independent of X_i, permitting us to write

$$E[T_iX_i] = E[(W_i + S_i)X_i] = E[W_iX_i] + E[S_i]E[X_i], \quad (14)$$

where $E[S_i] = 1/\mu$ and $E[X_i] = 1/\lambda$. Further, using iterated expectation and the exponential (λ) PDF of X_i, (13) implies

$$E[W_iX_i] = \int_0^\infty xE[W_i|X_i = x]f_{X_i}(x) \, dx \quad (15)$$

$$= \frac{\rho}{\mu^2(1-\rho)}. \quad (16)$$

From (7), (14) and (16), the average age can be obtained as

$$\Delta = \frac{1}{\mu}\left(1 + \frac{1}{\rho} + \frac{\rho^2}{1-\rho}\right). \quad (17)$$

For fixed service rate μ, we can minimize the average age Δ with respect to the arrival rate λ or, equivalently, the offered load $\rho = \lambda/\mu$. Differentiating (17) with respect to ρ and setting to 0, we obtain that the optimal utilization ρ^* satisfies the equation $\rho^4 - 2\rho^3 + \rho^2 - 2\rho + 1 = 0$ and thus $\rho^* \approx 0.53$. The server is idle $\approx 47\%$ of the time. The optimal age is achieved by choosing a λ that biases the server towards being busy only slightly more than being idle. At the optimal *utilization* ρ^*, the average number of packets in the system is $\rho^*/(1-\rho^*) \approx 1.13$.

Note that we would want ρ close to 1 if we wanted to maximize the throughput, which is the number of packets delivered to the monitors every second. If we instead wanted to minimize packet delay, that is minimize the system time of a packet, we would want ρ to be close to 0.

V. M/D/1 - FIRST COME FIRST SERVED

In certain systems, the service facility represents an aggregator of randomly arriving status updates in which the update packets are fixed length and packet processing times are deterministic. For example, this may describe a health care facility in which patients' heart rates updates are collected by a central monitor. Under the assumption that the generation of such information at the patients is independent and identically distributed, the aggregate traffic at the health care center may be modeled as a Poisson process of rate λ. This system can thus be abstracted as an $M/D/1$ system.

In a $M/D/1$ system, packet arrivals are Poisson with rate λ and service times is deterministic; $S_i = D$ for all i. We know that $E[X^2] = 2/\lambda^2$. To find the average age (7) we need to calculate $E[T_iX_i]$. We have $T_i = W_i + D$ and

$$E[T_iX_i] = E[W_iX_i] + DE[X_i]. \quad (18)$$

From equation (11), we have

$$E[W_i|X_i = x] = E[(T - x)^+] = E[(W + D - x)^+]$$
$$= E[W] + D - x$$
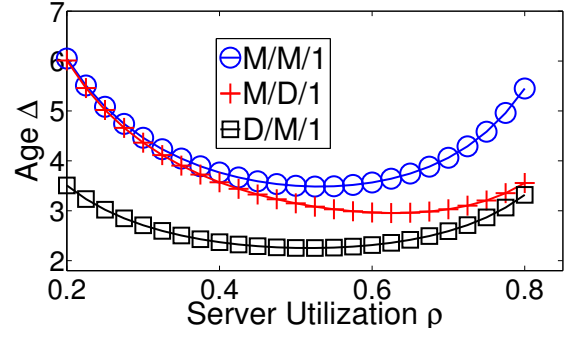$$- \mathcal{U}_D(x)\int_0^{x-D}(w + D - x)f_W(w) \, dw, \quad (19)$$

Fig. 3: An illustration of change of age (at a monitor) with server utilization for $M/M/1$, $M/D/1$ and $D/M/1$ systems using FCFS. The service rate assumed is $\mu = 1$.

where

$$\mathcal{U}_D(x) = \begin{cases} 1 & x \geq D, \\ 0 & \text{otherwise} \end{cases} \quad (20)$$

denotes the delayed unit step function. Note that the waiting time W of an $M/D/1$ system has expected value [14] $E[W] = D\rho/(2(1-\rho))$ and that the density $f_W(w)$ is obtained as the derivative of the cumulative distribution function (CDF) (see for example [15] [16])

$$F_W(w) = (1-\rho)\sum_{k=0}^{\lfloor w/D \rfloor} \rho^k\left(k - \frac{w}{D}\right)^k e^{\rho(\frac{w}{D}-k)}. \quad (21)$$

Using (19) for the conditional expectation $E[W_i|X_i = x]$, we calculate $E[W_iX_i]$, as given by (15), numerically. Having calculated $E[W_iX_i]$, we can calculate the age Δ using (7). We find that a utilization of $\rho \approx 0.625$ minimizes the age.

Figure 3 shows the variation of age with ρ for $M/M/1$ and $M/D/1$ FCFS systems for service rate $\mu = 1$. Note the similarity in age for smaller utilizations. This is because the arrival processes are the same and the age of status is more influenced by the small rate of status update packets. For larger utilizations, the server more often has a packet waiting for service and the age of status is influenced by the time a packet waits for service. For such ρ the $M/M/1$ system sees a larger age because packets with longer than average service times can lead to large waiting times for those that follow. It has a larger average system time than the deterministic system. It is also why a smaller age is achieved by the deterministic system, over all ρ.

VI. D/M/1 - FIRST COME FIRST SERVED

In a $D/M/1$ system the status packets are generated at a fixed period, say D, and the service times are exponentially distributed with mean $1/\mu$. For this system we have $\lambda = 1/D$, $E[X^2]/2 = D^2/2$ and $E[XT] = DE[T]$. Substituting in (7), we can write the expression for age as

$$\Delta = \frac{1}{D}\left[\frac{D^2}{2} + DE[T]\right]. \quad (22)$$

The average system time can be written as

$$E[T] = E[S] + E[W] = \frac{1}{\mu} + \frac{\beta}{\mu(1-\beta)}, \quad (23)$$

where $0 \leq \beta \leq 1$ is the solution of the equation $\beta = L_X(\mu(1-\beta))$, with $L_X(\cdot)$ being the Laplace transform of the distribution of inter-arrival times [14]. For deterministic arrivals with period D, the distribution is $f_X(x) = \delta(x-D)$, which gives $L_X(s) = e^{-sD}$. We can thus write β for the case of deterministic arrivals with period D as

$$\beta = e^{-\mu(1-\beta)D} = -\rho\mathcal{W}\left(-\rho^{-1}e^{(-1/\rho)}\right), \qquad (24)$$

where $\mathcal{W}(.)$ is the Lambert W function and the utilization $\rho = 1/(\mu D)$. From (22) and (23) the age can be obtained as

$$\Delta = \frac{1}{\mu}\left[\frac{1}{2\rho} + \frac{1}{1-\beta}\right]. \qquad (25)$$

We plot the age for a range of ρ in Figure 3. The age is minimized at a utilization $\rho = 0.515$. While the optimal utilization is almost as small as in the case of an $M/M/1$ system, the age achieved, shown in Figure 3, is smaller for $D/M/1$. At the optimal ρ, the memoryless arrivals lead to a 50% increase in age relative to deterministic arrivals.

VII. Minimum Achievable Age

For the FCFS systems we have analyzed up to now, the update generator (source) can neither observe nor control the state of the packet update queue, and the optimal load ρ^* strikes a balance between overloading the queue and leaving the queue idle. Next we derive lower bounds to the age by considering a system in which the update generator observes the state of the packet update queue so that a new status update is generated the very moment the previous update finishes service. In this setting, the server is always busy and the waiting time of every update packet is zero. Since each delivered update packet is as young as possible, the average status update age obtained for this system is a lower bound to the age for any queue in which updates are generated as a stochastic process independent of the current state of the queue.

Referring to the age function $\Delta(t)$ in Figure 2, $t_i = t'_{i-1}$ for all packets i in this system. For any packet i, $W_i = 0$. Also, $T_i = S_i + W_i = S_i$, $\lambda = 1/E[S]$ and $X_{i+1} = S_i$. Further, $E[XT] = E[X_iS_i] = E[X_i]E[S_i] = (E[S])^2$. The average age given in (7) can thus be written as

$$\Delta^* = \frac{1}{E[S]}\left[\frac{E[S^2]}{2} + (E[S])^2\right]. \qquad (26)$$

For a system with memoryless service at rate μ, the minimum average age is $\Delta^* = 2/\mu$.

However, non-negativity of the variance of S implies $E[S^2] \geq (E[S])^2$. Thus, for all service time distributions with $E[S] = 1/\mu$, (26) yields the lower bound

$$\Delta^* \geq \frac{3E[S]}{2} = \frac{3}{2\mu}. \qquad (27)$$

This lower bound is achieved by the system when the service times are deterministic.

VIII. Conclusions and Future Work

We have looked at the problem of keeping the status updates generated by a source as new as possible at interested recipients, given a set of physical constraints. We devised a method that helped us analyze such systems abstracted in queue-theoretic terms, where the source was modeled as an arrival process and the physical constraints as a given service facility. We looked at single source and server (and one or more monitors) systems under different assumptions of arrival and service processes and the queue discipline of first-come-first-served (FCFS). The arrival rate that minimizes the status update age for the considered FCFS systems is derived. It sets the utilization of the server to ensure that neither the packets wait too long for service nor the server is idle too often. Finally, we showed that the smallest age under FCFS can be achieved if a new packet is available exactly when the packet in service finishes service.

We plan to extend the work to other queue disciplines like last-come-first-served. The general case, when we have a network of sources connected to their monitors via different service facilities is also of interest.

References

[1] D. S. Alberts, D. S. Papp, and et al., *The Information Age: An Anthology on Its Impacts and Consequences (Information Age Anthology, Volume I); Part 1: The Information and Communication Revolution.* Natl Defense Univ Pr, Jun. 1997.

[2] A. Mainwaring, D. Culler, and et al., "Wireless sensor networks for habitat monitoring," ser. WSNA '02. ACM, 2002, pp. 88–97.

[3] P. Papadimitratos, A. La Fortelle, and et al., "Vehicular communication systems: Enabling technologies, applications, and future outlook on intelligent transportation," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 84–95, Nov. 2009.

[4] S. Kaul, M. Gruteser, V. Rai, and J. Kenney, "Minimizing age of information in vehicular networks," in *IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, 2011.

[5] S. Kaul, R. Yates, and M. Gruteser, "On piggybacking in vehicular networks," in *Global Telecommunications Conference, 2011. IEEE GLOBECOM 2011. IEEE*, Dec. 2011.

[6] A. Karakasidis, P. Vassiliadis, and E. Pitoura, "ETL queues for active data warehousing," in *Proceedings of the 2nd international workshop on Information quality in information systems*, ser. IQIS '05. Baltimore, Maryland: ACM, 2005, pp. 28–39, ACM ID: 1077509.

[7] H. Yu, L. Breslau, and S. Shenker, "A scalable web cache consistency architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 29, no. 4, pp. 163–174, Aug. 1999, ACM ID: 316219.

[8] M. Xiong and K. Ramamritham, "Deriving deadlines and periods for real-time update transactions," in *The 20th IEEE Real-Time Systems Symposium, 1999. Proceedings.* IEEE, 1999, pp. 32–43.

[9] Y. C. Hu and D. B. Johnson, "Ensuring cache freshness in on-demand ad hoc network routing protocols," in *Proceedings of the second ACM international workshop on Principles of mobile computing*, 2002.

[10] V. C. Giruka and M. Singhal, "Hello protocols for ad-hoc networks: overhead and accuracy tradeoffs," in *WoWMoM*. IEEE, Jun. 2005.

[11] T. He, Krishnamurthy, and et al., "Energy-efficient surveillance system using wireless sensor networks," ser. MobiSys '04. Boston, MA, USA: ACM, 2004, pp. 270–283, ACM ID: 990096.

[12] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Transactions on Mobile Computing*, vol. 1, no. 1, pp. 70–80, Mar. 2002.

[13] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes with Errata Sheet.* McGraw-Hill, Dec. 2001.

[14] R. Nelson, *Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling.* Springer-Verlag New York, Inc., 1995.

[15] A. Erlang, "The theory of probabilities and telephone conversations," *Nyt Tidsskrift for Matematik*, vol. 20, no. B, pp. 33–39, 1909.

[16] G. Franx, "A simple solution for the M/D/c waiting time distribution," *Operations Research Letters*, vol. 29, no. 5, pp. 221–229, Dec. 2001.