

# POWER-SPEED: A Power-Controlled Real-Time Data Transport Protocol for Wireless Sensor-Actuator Networks

Yangfan Zhou Edith C.-H. Ngai Michael R. Lyu  
Department of Computer Science and Engineering  
The Chinese University of Hong Kong  
Hong Kong, China  
Email: {yfzhou, chngai, lyu}@cse.cuhk.edu.hk

Jiangchuan Liu  
School of Computing Science  
Simon Fraser University  
Vancouver, Canada  
Email: jcliu@cs.sfu.ca

**Abstract**—This paper investigates the data transport problem for reporting delay-sensitive events in wireless sensor-actuator networks (WSANs). We specifically tailor the protocol design according to the features of WSANs and propose POWER-SPEED, a real-time data transport protocol for WSANs to achieve energy-efficient data transport for delay-sensitive event reporting. In POWER-SPEED, sensor nodes select the next-hop neighbor to actuators according to the spatio-temporal historic data of the upstream QoS condition, which completely avoids control packets. With an adaptive transmitter power control scheme, POWER-SPEED conveys packets in an energy-efficient manner while maintaining soft real-time packet transport. It thus reduces the energy consumption of data transport while ensuring the QoS requirement in timeliness domain. We demonstrate the effectiveness of POWER-SPEED through simulations with NS2.

## I. INTRODUCTION

In-situ sensing with wireless sensor networks (WSNs) has become a very promising technique with the advances in Micro Electro-Mechanical Systems (MEMS) and wireless communications [1]. In recent years, powerful devices, called actuators, have been introduced into WSNs. Actuators, (*e.g.*, robots) have more powerful computational capability, more energy resource, and they are mobile. An actuator can perform diverse tasks, such as processing the data reported from the sensors and accordingly interacting with the environment. The stationary sensors and the mobile actuators can form a powerful and yet cost-effective hybrid network, *i.e.*, a wireless sensor-actuator network (WSAN) [2].

There have been many novel potential applications of WSNs as well as WSANs, such as environmental monitoring and national-borders protection [2][3][4]. In WSANs, data collection on the physical phenomena of interest is still a major and critical work like that in WSNs. Usually, it is required that the actuators can react with the sensing events timely and accordingly perform interactions with the environment. For example, if the sensors in a WSAN that monitors a forest-beat report abnormal temperature in some area, the actuators should move to that area to check whether there is a fire in a timely manner. QoS guarantee in the domain of timeliness is highly desired in the sensor-to-actuator data transport protocol

for delay-sensitive event-reporting of WSANs.

Unfortunately, like in WSNs, sensors in WSANs are still inherently subject to limited energy resource as recharging the batteries of sensors is impractical due to the unattended operational features of the network [2]. Therefore, the main challenge of designing a real-time data transport protocol for delay-sensitive event-reporting of WSANs is to achieve energy-efficiency. The protocol should guarantee that the actuators can receive packets on delay-sensitive events in given latency bounds and the transport scheme should cost as low energy as possible.

There are many existing real-time data transport protocols in the literature for WSNs. Lu *et al* [5] described a packet scheduling policy called Velocity Monotonic Scheduling. It accounts for both time and distance constraints. SPEED [6] by He *et al* combines feedback control and non-deterministic QoS-aware geographic forwarding. Felemban *et al* [7] proposed Multi-path and Multi-Speed Routing Protocol (MM-SPEED) for probabilistic QoS guarantee in WSNs. Multiple QoS levels are provided in timeliness domain by using different delivery speeds, while various requirements are supported by probabilistic multipath forwarding in reliability domain.

However, the network and traffic features of WSANs are different from those of WSNs. In WSANs, actuators are mobile sinks of the data traffic. The mobility of data sinks poses that the traffic routes must be reestablished frequently. The transport protocol for WSANs should be stateless. Also, actuators can directly communicate with each other as they can be equipped with powerful antennae. Therefore, sensor reporting packets can be delivered in an *anycast* manner. The transport protocol for WSANs can convey packets to any of the actuators. In this paper, we specifically tailor the solution of the real-time data transport problem according to these features of WSANs. We address this challenging problem by proposing POWER-SPEED, a real-time data transport protocol for WSANs to achieve energy-efficient data transport for delay-sensitive event reporting. With POWER-SPEED, a sensor node can estimate the downstream QoS condition to actuators based on the spatio-temporal historic data of the upstream paths.

Without requiring any control packets, sensor nodes can select the next-hop neighbor to actuators according to the estimation. They then efficiently adjust the power level of its wireless transmitter to a minimum value under the constraint that the packet sent by this node could just reach its intended neighboring node. In this way, POWER-SPEED reduces the energy consumption of data transport while maintaining QoS requirement in timeliness domain.

The rest of the paper is organized as follows. In Section II, we discuss the real-time data transport problem according to the network and application features of WSANs. Section III elaborates our real-time data transport protocol, namely, POWER-SPEED. In Section IV, we present our simulation results. We provide conclusion remarks in Section V.

## II. ENERGY-EFFICIENT REAL-TIME DATA REPORTING OF DELAY-SENSITIVE EVENTS

We consider a WSAN that consists of a collection of sensor nodes  $S$  and actuator nodes  $A$ . Each sensor node is denoted by  $s_i$  ( $s_i \in S$ ,  $i = 1, 2, \dots, |S|$ ). Sensor nodes are stationary. Their locations do not change after they are deployed. Each actuator node is denoted by  $a_j$  ( $a_j \in A$ ,  $j = 1, 2, \dots, |A|$ ). Actuator nodes are mobile.

For an event that takes place in the network area, a subset  $R$  of  $S$  can sense the event and report event data to the actuators. We call these nodes *reporting nodes*, each of which is denoted by  $r_k$  ( $r_k \in R$ ,  $k = 1, 2, \dots, |R|$ ). Because the actuators can directly communicate with each other as they can be equipped with powerful antennae, we consider that the destination of the reporting packets can be any of the actuators, *i.e.*, reporting traffic can be routed in an anycast manner to the actuators.

For each in-network node  $u$ ,  $N(u)$  denotes the set of nodes which node  $u$  can communicate directly without relaying of other nodes. We call nodes in  $N(u)$  the *neighbors* of node  $u$ . Each node is aware of the existence of each of its neighbors.

### A. Location-aware networks

WSANs are employed to sense and handle environmental data. It is usually required that each in-network node is aware of its geographic location as the location information is necessary to identify and locate physical phenomena. In such WSANs, each sensor or actuator node should know its approximate geographic location. Location information is achievable if each node carries a GPS receiver or if some localization algorithms (*e.g.*, that in [8]) are employed.

Such location information can greatly facilitate the design of a data transport protocol, which will be explored in our following discussion. The location of a node is denoted by  $X(u)$ . We define  $Dis(u, v)$  as the physical (Euclidian) distance between node  $u$  and node  $v$ ,

$$Dis(u, v) = \|X(v) - X(u)\|, \quad (1)$$

where  $u$  and  $v$  can be  $s_i$  or  $a_i$ .

### B. Stateless data transport via dynamic paths

In WSANs, the actuators, *i.e.*, the sinks of sensor-reporting traffic are mobile. As a result, the network topology changes frequently. Global shortest-path routing from reporting nodes to the actuators is not feasible because frequent reestablishment of shortest paths inevitably causes high overhead in terms of energy required for exchanging control packets. Therefore, to be energy-efficient, the sensor reporting packets have to be forwarded in a stateless manner. An in-network node does not maintain a routing table to the actuators. It should find out where to forward the sensor reporting packets by exchanging as few control packets as possible with its neighbors. The data transport protocol must convey sensor reporting packets to the actuators via dynamic paths in order to adapt to the mobility of the actuators.

### C. Delay-sensitive data transport

In most application cases of WSANs, reporting data are delay-sensitive. For example, in a WSAN that performs real-time environmental monitoring, it is required that environmental data are obtained by the in-network actuators within a predefined latency bound. Different events may have different latency-bound requirements.  $L(e)$  denotes the latency bound of event  $e$ . We consider a packet on a particular event from a reporting node is successfully delivered if the packet can reach an actuator with latency less than the latency bound. A protocol for delay-sensitive data transport should successfully deliver as many packets as possible. It should also adapt to the different latency-bound requirements of different events.

### D. Power-controlled packet transmission

Since WSANs are usually location-aware networks, we notice that such location information can be utilized to conduct power control of a node's wireless transmitter. A transmitter power control scheme enables each node to set its power level to a minimum value under the constraint that the packet sent by this node could just reach its intended neighbor. The energy consumption of data transport can thus be reduced. Power-controlled packet transmission is an important technique to save the energy consumptions of sensor nodes and prolong the lifetime of a network.

The prerequisite of a transmitter power setting scheme is that each sensor node can set its own wireless transmitter power level. This is true in typical sensor node implementations. For example, the Berkeley Mica Mote [9] provides such program interfaces.

According to the wireless signal fading model [10], packets transmitted from node  $u$  can reach its intended neighbor  $v$  if the transmitter power setting of node  $u$  satisfies:

$$Pr(u) \geq c \cdot (Dis(u, v))^n \quad (2)$$

Here  $c$  is a constant related to the wireless system parameters.  $n$  is the signal fading factor whose value is typically in the interval [2, 5]. The optimal transmitter power setting for node  $u$  to send a packet to node  $v$  is therefore computed by:

$$Pr(u) = c \cdot (Dis(u, v))^n = c \cdot \|X(v) - X(u)\|^n \quad (3)$$

### III. DESIGNING THE POWER-SPEED PROTOCOL

#### A. Estimating the hop-by-hop delays

As a protocol designed to transport real-time packets on delay-sensitive events, POWER-SPEED requires to know how long a packet has been experienced in the network. As each data packet has a *deadline* (i.e., the maximum time the packet can experience before it reaches an actuator), the remaining time before the packet expires is a crucial parameter for POWER-SPEED to select where to forward the packet and thus guarantee that the packet can reach an actuator before expiration. Before discussing the details on this next-hop selection scheme in Section III-C, we elaborate the mechanism we adopt to calculate the delay a packet has experienced and how to estimate traffic condition of the downstream path (i.e., hop-by-hop delay a packet will experience in the future).

To facilitate the following discussion, we demonstrate part of a POWER-SPEED data packet in Figure 1.

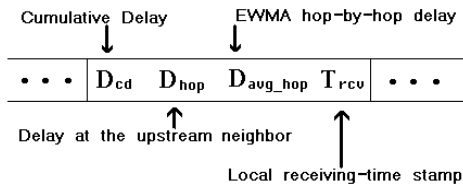


Fig. 1. Part of a POWER-SPEED packet

The interval between the time when node  $u$  receives a packet and the time the packet is sent and received by node  $u$ 's intended neighbor  $v$ , denoted by  $Delay(u)$  is:

$$Delay(u) = t_{prop} + t_{proc} + t_q + t_{tran} \quad (4)$$

where  $t_{prop}$  is the propagation delay from  $u$  to its intended neighbor,  $t_{proc}$  is the processing delay at node  $u$ ,  $t_q$  is the queuing delay at node  $u$  during which the packet is waiting to be sent out, and  $t_{tran}$  is the transmission delay which is related to the channel bandwidth and packet size. We call  $Delay(u)$  the *hop-by-hop delay* between node  $u$  and its neighbor.

In POWER-SPEED protocol, node  $u$  timestamps the  $T_{rcv}$  field of a packet when the packet is received. When it sends the packet, it can encapsulate the time experienced by the packet inside node  $u$  into the  $D_{hop}$  field of the packet. This time value is  $t_{proc} + t_q$  in Equation (4).  $t_{prop}$  is the light-speed propagation delay which can be ignored.  $t_{tran}$  can be calculated by its intended node (i.e., node  $v$ ) by bandwidth value and packet size. With  $t_{tran}$  and the encapsulated value of  $D_{hop}$ , node  $v$  can thus approximately calculate  $Delay(u)$ . Node  $v$  can calculate the *cumulative delay* (denoted by  $D_{cd}$ ) that the packet has experienced in the network when it receives the packet and update the  $D_{cd}$  field of the packet (which initially is zero when the packet is generated) by adding  $Delay(u)$ .

The current  $Delay(u)$  is also averaged by node  $v$  with an exponentially weighted moving average (EWMA) approach [11] to get the estimation of hop-by-hop delay from node  $u$

to node  $v$ .  $\tilde{D}(u)$  denotes this estimation. Specifically,  $\tilde{D}(u)$  is updated with current  $Delay(u)$  according to:

$$\tilde{D}(u) = \alpha \tilde{D}(u) + (1 - \alpha) Delay(u) \quad (5)$$

where  $\alpha$  is a constant.

To describe how an intermediate relaying node  $s_n$  estimate the future hop-by-hop delay a packet will experience, denote  $\{r_1, s_1, s_2, \dots, s_n\}$  as the traffic path from a source node  $r_1$  to  $s_n$ . In POWER-SPEED, node  $s_n$  estimates the future hop-by-hop delay a packet will experience based on only the hop-by-hop delay estimation of the upstream hops the packet has traveled, i.e., the estimation of the hop-by-hop delay between each pair of adjacent nodes in  $\{r_1, s_1, s_2, \dots, s_n\}$ . This means that POWER-SPEED does not require feedbacks from downstream nodes to estimate the future hop-by-hop delay a packet will experience. It thus completely avoids control packets, which featuring one of the merits of POWER-SPEED. Details are as follows.

Node  $s_1$  can know  $\tilde{D}(r_1)$  as discussed above. It then fills the  $D_{avg\_hop}$  field of the data packets with  $\tilde{D}(r_1)$ . For node  $s_i$  ( $1 < i \leq n$ ), it updates the  $D_{avg\_hop}$  field according to:

$$D_{avg\_hop} = \beta D_{avg\_hop} + (1 - \beta) \tilde{D}(s_{i-1}) \quad (6)$$

where  $\beta$  is a constant. Because  $\tilde{D}(s_{i-1})$  is calculated at node  $s_i$  and previous  $D_{avg\_hop}$  is obtainable from packets, no control packet is required to obtain  $D_{avg\_hop}$  according to Equation 6.

In POWER-SPEED, node  $s_n$  deems that the  $D_{avg\_hop}$  calculated in Equation 6 as the estimation of the future hop-by-hop delay the packet will experience in its future journey. Such an estimation approach is very reasonable because it employs a 2-dimensional (spatio-temporal) EWMA estimation (i.e., Equations (5) and (6)) that considers the impact of both time and space historical data.

#### B. Calculating packet transport speed and packet forwarding candidates

In POWER-SPEED, packet transport *speed* is defined as the average hops a packet can go through in one second. With the notations used in Section III-A, node  $s_n$  calculates the packet speed by:

$$speed = \frac{1}{D_{avg\_hop}} \quad (7)$$

where  $D_{avg\_hop}$  is obtained with Equation (6). Note that *speed* is a positive real number.

Node  $s_n$  can calculate the maximum number of hops the packet can go through before its expiration with:

$$max\_hops = speed \cdot (deadline - D_{cd}) \quad (8)$$

For each actuator  $a_j$ ,  $s_n$  calculates whether each node  $w$  in  $N(s_n)$  is nearer than  $s_n$  to  $a_j$ <sup>1</sup>. If this is true,  $s_n$  estimate the

<sup>1</sup>This is feasible as a sensor node can easily know the locations of its neighbors. Actuators, although mobile, can broadcast their current locations to sensor nodes as they are equipped with powerful antennae

number of hops a packet has to travel from  $s_n$  via  $w$  to  $a_j$  by Equation (9). If  $hops < max\_hops$ , node  $w$  is considered as one of the packet forwarding candidates. It means that the packet can go to actuator  $a_j$  via node  $w$  and it is expected to arrive at actuator  $a_j$  before it expires.

$$hops = \frac{Dis(s_n, a_j)}{Dis(s_n, a_j) - Dis(w, a_j)} \quad (9)$$

### C. Selecting next-hop neighbor

To be energy-efficient, a node (also denoted by  $s_n$ ) should send packets to one of its packet forwarding candidates so that the total energy consumption required to deliver the packet from the node to an actuator is considered minimized. That is to say, node  $s_n$  should locally minimize the total energy consumption required for a packet to reach an actuator.

Because the energy consumption for receiving a packet and processing a packet is constant, we consider the energy required for sending a packet to each packet forward candidate  $w$  for node  $s_n$ , denoted by  $E(s_n, w)$ . With transmitter power control, it is as follows.

$$E(s_n, w) = \gamma Pr(s_n) = \gamma c \cdot (Dis(s_n, w))^n \quad (10)$$

where  $\gamma$  is a constant related to the packet size.

Node  $s_n$  estimates the total energy consumption for the packet to reach actuator  $a_j$  via node  $w$  as follows.

$$total\_e(s_n, w) = hops \cdot E(s_n, w) \quad (11)$$

where  $hops$  is calculated with Equation (9).

Among all packet forwarding candidates, node  $s_n$  selects the node that achieves the minimum value of total energy consumption calculated with Equation (11). It then sets its transmitter power level according to Equation (3) and forwards the packet to this candidate. In such a next-hop selection scheme, a packet that will expire in a longer period of time will adaptively be sent with lower transmitter power level to save energy. On the contrary, a packet that will expire sooner will be sent adaptively with higher transmitter power level, which results in fewer hop numbers between the sender to a destination actuator, and thus guarantees that the packet can reach its destination in a shorter period of time.

Note that this next-hop selection scheme in POWER-SPEED requires no control packets. And it is a fully distributed and localized algorithm which well suits WSANs.

## IV. SIMULATION RESULTS

We program POWER-SPEED with the NS-2 [12] simulator and study the performance of POWER-SPEED with simulations. We investigate the performance of POWER-SPEED in terms of the total energy consumption to report data on delay-sensitive events and the proportion of the packets that can be delivered to actuators in time. For comparison, we also do simulations with the geographic routing protocol (GRP) [13], where greedy forwarding is employed and no control packet is required like POWER-SPEED. Detailed settings of the simulation network are as follows.

- Area of Sensor Field: 200m\*200m
- MAC Protocol: IEEE 802.11 without RTS/CTS and ACK
- Bandwidth: 2 Mbps
- Maximum Wireless Communication Range: 40m
- Wireless Communication Model: Two-Ray Ground
- Packet Size: 36 bytes

In order to have a direct observation on the energy-saving effect of POWER-SPEED, we first study the protocol performance in case that there is only one concurrent event in the network. We randomly deploy  $i$  sensor nodes in a uniform manner. Let the event takes place at location (50m, 50m). The reporting of the event by source nodes lasts for 1000 seconds. We change  $i$  from 100 to 250, and we change the reporting rate of source node from 5 to 30 packets/second. For each setting, we select different random seeds and run simulations for 5 times. Results are averaged.

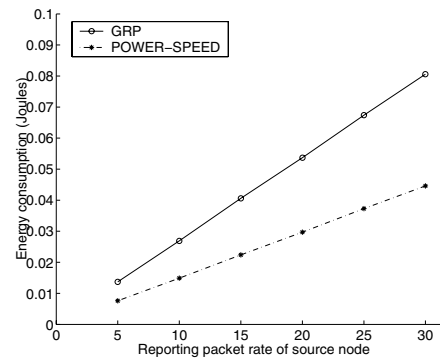


Fig. 2. Total energy consumptions of different source reporting rates

Figure 2 shows the total energy consumption of the whole network in case that  $i = 100$  when different report rates of source node are set. The curves are in linear manner as energy consumption is linearly related to number of packets transmitted. It can be seen that POWER-SPEED effectively reduces the total energy consumption for conducting an event-reporting task. It saves about 40% energy comparing to the GRP protocol. This is not surprising due to the energy-saving technique adopted in POWER-SPEED.

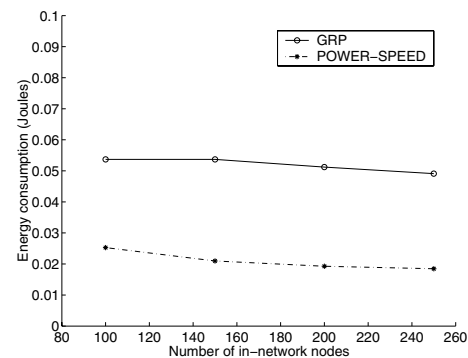


Fig. 3. Total energy consumptions of different node number

Figure 3 further shows the impact on in-network node number on the total energy consumption in case that the reporting rate of source node is 20 packets/s. We can see that when the

node density increases, the enhancement of POWER-SPEED is better than that of GRP. This is natural. GRP adopts a best-effort packet-forwarding mechanism with fix transmission range, the hop numbers between sources and actuators are nearly the same although node density increases. POWER-SPEED, on the other hand, adapts to node density and selects more energy-efficient paths as node density increases.

To study how well POWER-SPEED adapts to different latency requirements, we conduct simulations in which two concurrent delay-sensitive events with different latency requirements coexist. We randomly deploy 100 sensor nodes in a uniform manner. We let the event  $e_1$  and  $e_2$  takes place at locations  $(50m, 50m)$  and  $(80m, 80m)$ . The reporting of the event by source nodes lasts for 1000 seconds. The maximum tolerable delays of packets on  $e_1$  and  $e_2$  are  $0.025s$  and  $1s$  in order to investigate how POWER-SPEED performs in reporting a more critical (in term of latency bound requirement) but farther event  $e_1$ . We set the reporting rate of source node in the range  $(10, 30)$ . For each setting, we also set different random seeds to run simulations, the results of which are averaged.

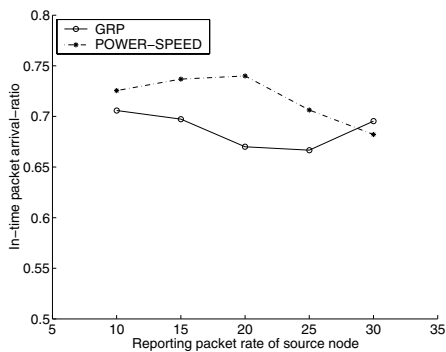


Fig. 4. Comparison of in-time packet arrival-rates

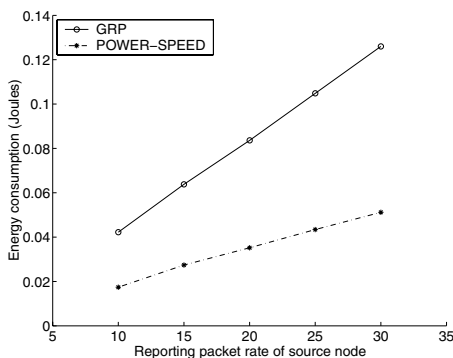


Fig. 5. Comparison of total energy consumptions

Figure 4 demonstrates the in-time packet arrival-rate of packets on  $e_1$  and Figure 5 demonstrates the total energy consumption. POWER-SPEED is comparable to GRP in terms of in-time packet arrival-rate of a critical event (*i.e.*,  $e_1$ ). But it saves more than 50% of energy comparing to GRP. This is because POWER-SPEED can adapt well to events with different latency requirements. Packets on a less critical event

(*i.e.*,  $e_2$ ) can be sent with lower transmitter power level to save energy, while packets on a more critical event (*i.e.*,  $e_1$ ) can be sent with higher transmitter power level to guarantee the arrival-rate.

## V. CONCLUSIONS

This paper studies the problem of real-time data transport for reporting delay-sensitive events in WSANs. We carefully tailor our protocol design by examining the features of WSANs, and propose POWER-SPEED, a power-controlled real-time data transport protocol for WSANs. POWER-SPEED avoids the overhead of control packets by estimating the downstream path quality (in terms of delay) with only spatio-temporal historic data of the upstream path quality, which is obtainable without exchanging additional packets. POWER-SPEED selects the next-hop neighbor based on the downstream path quality and the latency-bound requirement of packets. Adaptively, it sends a packet that will expire in a longer period of time with lower transmitter power level to save energy. And it sends a packet that will expire sooner with higher transmitter power level, which results in fewer hop numbers between senders to destination actuators, and thus guarantees that the packet can reach its destination in a shorter period of time. In this way, POWER-SPEED achieves energy-efficiency while maintaining the QoS requirement in timeliness domain. We perform extensive simulations with the popular network simulation tool NS2 and the results demonstrate the effectiveness of our protocol.

## REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on wireless sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, 2002.
- [2] I. Akyildiz and I. Kasimoglu, "Wireless sensor and actor networks: research challenges," *Elsevier Ad Hoc Networks Journal*, October 2004.
- [3] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless sensor networks for habitat monitoring," in *Proc. of the ACM WSNA*, 2002.
- [4] N. Xu, S. Rangwala, K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, "A wireless sensor network for structural monitoring," in *Proc. of the ACM SenSys*, Baltimore, MD, November 2004.
- [5] C. Lu, B. Blum, T. Abdelzaher, J. Stankovic, and T. He, "RAP: A real-time communication architecture for largescale wireless sensor networks," in *Proc. of the IEEE RTAS*, San Jose, CA, September 2002.
- [6] T. He, J. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A realtime routing protocol for sensor networks," in *Proc. of the IEEE ICDCS*, Providence, RI, May 2003, pp. 46–55.
- [7] E. Felemban, C.-G. Lee, E. Ekici, R. Boder, and S. Vural, "Probabilistic QoS guarantee in reliability and timeliness domains in wireless sensor networks," in *Proc. of the IEEE Infocom*, Miami, FL, U.S., March 2005.
- [8] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low-cost outdoor localization for very small devices," *IEEE Personal Communication*, October 2000.
- [9] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Proc. of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, Cambridge, MA, November 2000.
- [10] T. Rappaport, *Wireless Communications: Principles and Practices (2nd Edition)*. Upper Saddle River: Prentice Hall, 2002.
- [11] J. S. Hunter, "The exponentially weighted moving average," *Journal of Quality Technology*, vol. 18, pp. 203–210, 1986.
- [12] K. Fall and K. Varadhan, *The ns manual*, <http://www.isi.edu/nsnam/ns>.
- [13] B. Karp and H. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proc. of the 6th MobiCom*, Boston, MA, August 2000.