Power-aware QoS Geographical Routing for Wireless Sensor Networks - Implementation using Contiki

Djamel Djenouri[1,2], Ilangko Balasingham[2,3]
CERIST Research Center, Algiers, Algeria,
[2] Department of Electronics and Telecommunications, NTNU, Trondheim, Norway
[3] Interventional Center, Oslo Univetrsity Hospital, Rikshospitalet HF, Oslo, Norway
ddjenouri@mail.cerist.dz, ilangko.balasingham@medisin.uio.no

***Abstract*— This paper presents the design and implementation of a new geographical quality of service (QoS) routing for wireless sensor networks. The protocol is based on traffic differentiation and provides customized QoS according to the traffic requirement. For each packet, the protocol attempts to fulfill the required data-related QoS metric(s) while considering power-efficiency. The data related metrics include packet latency and reliability, while power-efficiency has been considered for both power transmission minimization and residual energy maximization (load balancing). The protocol has been implemented in real sensor motes using Contiki operating system, which offers many modules and has many features that facilitate efficient communication protocol implementation. The protocol was then evaluated in a testbed. The experimental results show good QoS performance, and particularly, traffic-differentiation QoS as expected, i.e., QoS-sensitive packets were routed with better performances than regular packets. The protocol is generic and applies to any application with traffic requiring different QoS, such as in biomedical and vehicular applications.**

I. Introduction

We consider wireless sensor network applications where the sensed data are heterogeneous in terms of quality of service (QoS) requirements, for which we propose a new QoS routing. The proposed protocol [1] takes into account this feature and ensures traffic-based differentiation QoS. It is generic and may be applied to any application that has heterogonous traffic, such as vehicular and biomedical wireless sensor networks (WSN). The protocol is also localized and uses geographical information to select the next router. Unlike most of the protocols proposed in the literature that are evaluated merely by simulation, the proposed protocol was implemented in real sensor motes and evaluated using a WSN testbed. We consider a general scenario where sensor motes are deployed in the monitored environment and report the collected data either periodically or in realtime. The collected data are transmitted towards fixed sinks via other sensors in a multi-hop, ad hoc paradigm. Two kinds of sinks are used; primary sink and Secondary sink; secondary sink receives a separate copy of each message that requires high reliability. The two sinks must be placed in separate areas to avoid traffic congestion. Three different requirements are considered, i) energy efficiency, ii) packet delivery reliability, and iii) latency. Giving these requirements data traffic is classified into: i) regular traffic, ii) reliability-sensitive traffic, which should be delivered without loss but can tolerate reasonable delay, e.g., file transfer iii) delay-sensitive traffic, which should be delivered within a deadline but may tolerate reasonable packet loss, e.g., video streaming, and finally iv) critical traffic of high importance, requiring the highest reliability and delivery within a deadline. Following this classification the proposed protocol tries to ensure exactly the required QoS for each packet. Therefore, from routes supposed to ensure QoS requirements, the most energy efficient is selected.

The remaining of the paper is organized as follows. The next section presents the related work, followed by a description of the protocol in Section III. Section IV describes the implementation of the protocol in real sensor motes, while Section V shows some empirical results using a testbed of the sensor motes. Final, Section VI draws the conclusions.

II. Related Work

The use of geographical information in routing is a promising approach that makes the protocol localized, i.e. each node does not need any information beyond its one hop vicinity to make the routing decision, which enables high scalability. The route selection approach differs from a protocol to the other depending on the metric(s) to consider. In this paper route selection is based on quality of service (QoS) objectives. QoS routing using geographical information for WSN has been lately considered by many researchers, and new protocols have been proposed, such as SPEED [2], MMSPEED [3], DARA [4], GREES [5], RPAR [6], DHGR [7], and EAGFS [8]. Still, none of these protocols makes a clear differentiation between traffic in route selection with respect to QoS requirements. They define either the same combined metric (of all the considered QoS metrics) [2], [8], [5], or several services but with respect to only one metric [6], [3]. This may not be enough for some applications, such as vehicular and biomedical WSN, where different traffic may have different QoS requirements. The main contribution of the proposed protocol is to provide different QoS services with respect to both latency and reliability according to the traffic type, while simultaneously considering residual energy and transmission power. Unlike most of the protocols that were evaluated only by simulation, the proposed protocol has been implemented and evaluated using real sensor motes. A sensor mote consists of sensors, micro controller and wireless radio unit. A variety of sensor motes have been developed and used in many research and industrial projects. Crossbow [1] provides a series of motes with different features and capacities, which are widely used. To program and operate these sensor motes many operating systems have been designed and developed, such as TinyOS [9], MagnetOS [10], EmStar [11], Mantis [12], and Contiki [13]. The later is the first platform that supports TCP/IP, which is important for real life deployment in many applications. It facilitates the integration of sensor networks with other networks such as Internet. This system has many other interesting features and has therefore been selected for our implementation.

III. Protocol Overview

A. Assumptions

Each node is supposed to be aware of its own coordinates, which can be obtained using some distributed localization service. This position serves as the network (global) address. In addition, the node should be aware of its current battery state B_{v_i} (also termed *residual energy*). We assume that nodes have the same spherical transmission power range P_{range}, and that each node can control its transmission power. The set of nodes in v_i's vicinity denoted by N_{v_i} is called v_i's neighboring nodes, defined as $N_{v_i} = \{v_j : dist_{v_i,v_j} \leq P_{range}\}$, where $distv_i, v_j$ denotes the euclidian distance. In addition to N_{v_i}, we define the set of neighboring nodes providing positive advance from node v_i towards a final destination v_d, denoted by $N^{advc}_{v_i,v_d}$, as the set of neighboring nodes that are closer to the destination than v_i. It is given as $N^{advc}_{v_i,v_d} = \{v_j \in N_{v_i} : dist_{v_j,v_d} \leq dist_{v_i,v_d}\}$. Like all geographic routing protocols, each node needs to know about the positions of its neighboring nodes as well as the destination. A BEACON protocol is executed between neighboring nodes allowing mutual update of the neighboring node list, neighboring nodes' positions, and several other parameters [3], [5]. Neighbor node positions may change due to nodes' mobility. Further, the mobility may breaks links and creates others, resulting in changes in the neighboring list. The frequency of Beacon packet exchange must adapt to the degree of mobility. Node density is supposed to be high enough to prevent void situation. Void situation occurs when a router cannot find a node closer to the destination amongst its neighboring nodes.

B. Routing Regular Packets

The only metric considered when routing regular packets is energy. Both transmission power and residual energy of potential routers should be considered to achieve power efficiency [14]. To cope with this trade-off, a non-aggregated min-max approach has been used. The problem for regular packets is to select at node v_i the most power efficient node for destination v_d, from a set of neighboring nodes offering positive advance, $N^{advc}_{v_i,v_d}$. For each candidate, the standard deviation from its optimum is calculated. These deviations are denoted $Z_T(x)$ and $Z_B(x)$, for transmission power and residual energy respectively. Note that the optimum for the transmission power is the closest node (when assuming a free space propagation model), and the one for the residual energy is that having the most charged battery. The node whose maximum deviation is the minimal, say S_0, is selected. That is,

$$S_0 = \{x : \max_{m\in\{T,R\}} \{Z_m(x)\} = \min_{j\in N^{advc}_{v_i,v_d}} \max_{k\in\{T,R\}} \{Z_k(v_j)\}\}. \quad (1)$$

C. Routing Delay-sensitive Packets

Delay-sensitive packets are routed through routes assumed to meet the required deadline, while considering power-efficiency as the second objective. Assume delay-sensitive packets have a delivery deadline, dd, specified by the upper layers. It indicates the time the packet should be delivered to the sink node. For every node v_j in $N^{advc}_{v_i,v_d}$ two velocities are used; required velocity (speed), s_{req}, and offered velocity, s_{v_j}. Upon receiving a packet the recipient node stamps the corresponding reception time locally. To account for all the possible delays at the node, i.e., queuing, contention, retransmission, etc., it updates the deadline *prior to each transmission* in the MAC layer to account for the delay from receiving the packet until it reaches its final transmission. If the reception time is denoted by t_{rec}, the time of last transmission by t_{tr}, the bandwidth by bw, and the packet size by $size$, then the time remaining to the deadline, rt, is updated at node v_i as

$$rt = rt_{req} - (t_{tr} - t_{rec} + size/bw), \quad (2)$$

where rt_{req} is the value of rt at reception time, and $t_{tr} - t_{rec} + size/bw$ gives the entire delay from the reception of the packet at noe v_i until the transmission of the last bit. Upon reception of the packet at node v_i, the required speed is calculated using both the remaining time to the deadline and the remaining distance to the destination as given by

$$s_{req} = \frac{dist_{v_i,v_d}}{rt}. \quad (3)$$

For every candidate v_j, the offered velocity is estimated. Exponential Weighted Moving Average (EWMA) [15] is used to estimate waiting time at the queue for node v_i, say w_{v_i}, as well as to estimate transmission time to the next node, dtr_{v_j}, and waiting time at the queue of the latter, w_{v_j}. Given these estimates, the estimated velocity for node v_j is given as

$$s_{v_j} = \frac{dist_{v_i,v_d} - dist_{v_j,v_d}}{w_{v_i} + dtr_{v_j} + w_{v_j}}. \quad (4)$$

After computing velocities of all candidate nodes, node v_i calculates the set of nodes supposed to meet the required deadline, $N^{s_{req}}_{v_i,v_d}$, as

$$N^{s_{req}}_{v_i,v_d} = \{v_j \in N^{adv}_{v_i,v_d} : s_{v_j} \geq s_{req}\}. \quad (5)$$

This set is then used to extract the most power-efficient node, following the same procedure as with regular packets.

[1] www.xbow.com

D. Routing Reliability-Sensitive Packets

Reliability is addressed by sending a copy to both primary and secondary sinks. This multi-sink single-path approach is odopted instead of the single-sink multi-path approach used in [3], which results in data packets convergence near or at the sink and thus increases traffic contention and collisions. For each copy, the reliability module selects from $N^{advc}_{v_i,v_d}$ the node providing the highest packet reception ratio (prr). prr_j is estimated for each candidate v_j. It indicates the probability of successful delivery to node v_j. MAC ACKs are used as indication of recpetion/loss of packets at the next hop, and the estimation is updated with EWMA [15]. If more than one node provide the maximum value, then the most energy efficient is selected.

E. Routing Critical Packets

Both approaches used for reliability-sensitive packets and delay-sensitive packets are combined for critical packets. Duplicate packets are sent to each sink. For every copy, the most reliable node is sleeted among nodes supposed to ensure the required deadline (nodes of set $N^{s_{req}}_{v_i,v_d}$).

IV. Protocol Implementation

For the implementation of the proposed protocol TelosB[2] motes were used, along with Contiki operating system [13]. TelosB (TPR2420) has a USB port, which facilitates programming and data collection. It includes an IEEE 802.15.4 radio (CC2420) with integrated antenna. It has low power consumption and uses two AA batteries. This radio is a ZigBee compliant RF transceiver that operates in the industrial scientific and medical (ISM) band, at 2.4 to 2.4834 GHz. TelosB's microcontroller is an 8 MHz TI MSP430, with a 10 KB RAM and 1MB of external flash memory. The mote also includes integrated light, temperature, and humidity sensor suite (TPR2420). All these features make TelosB an appropriate research platform suitable for our experiments.

Contiki is written in C programming language and uses the protothread programming paradigm, which makes the coding easier compared to pure event-based systems [16]. It includes a variety of modules and libraries that facilitate the implementation of protocols and applications. We used RIME, a lightweight communication stack that provides a useful programming interface for communication protocol implementation. Unicast, broadcast, and multi-hop modules of RIME were used all together in the implementation. The unicast module enables to open a unicast virtual channel between a pair of nodes. Each node should open the same channel identified by a unique number (ID), and then it can transmit/receive packets using library/prototyped functions. This channel was used by the protocol to transmit/receive ACK packets upon data packet reception. This ACK mechanisms was implemented by the protocol since the current Contiki's MAC protocols do not use any ACK mechanism. EWMA estimation was implemented within packet reception function of the unicast channel, which was defined at the $unicast\_callbacks$ structure. The broadcast module was used to implement the BEACON protocol. The same steps are defined as with the unicast channel, i.e. opening the channel using a unique ID, defining the packet reception function at $broadcast\_callbacks$ structure, and using a library function to send packets. We did not use the announcement module available in Contiki as it needs an explicit reception, which requires synchronization and is not appropriate for the BEACON protocol implementation.

[2]www.xbow.com

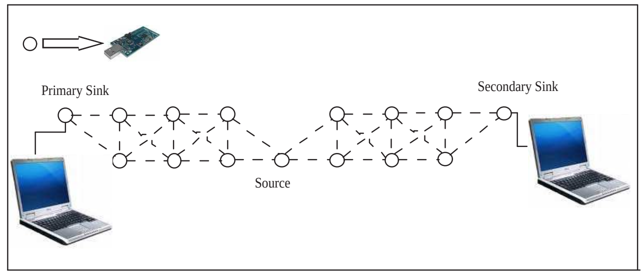

Fig. 1. Experimental Network Topology

The multi-hop module is the basic framework to implement any routing protocol. It provides a set of library functions facilitating the protocol integration into the system. Basically, two key prototyped functions were implemented and added to the $multihop\_callbacks$ structure. The first one is the code executed when receiving a packet as the final destination, while the second one is executed when receiving a packet as intermediate node (forward). Function calls are ensured by the multihop module, which releases the protocol from handling such events (packet receptions). In addition to the process implementing the protocol, another process is launched at run time to generate traffic. In the current experimental implementation, traffic generation process blocks and waits until the user button has been activated. This way, any sensor may generate traffic upon pressing its button. Traffic is then generated with a rate of 1 packet of 20 bytes per second. Traffic type was distributed uniformly with the following rates: 0.4 for regular-packets, and 0.2 for each of delay-sensitive, reliability-sensitive, critical packets. In real-world implementation, any traffic generation process may be implemented, which usually depends on the application and sensor reading. The size of the whole binary code was 24 KB. This includes code for routing protocol, traffic generation process, the operating system kernel. The whole code uploads into the mote as a single file.

V. Experimental Results

Fifteen TelosB motes uploaded with the implementation described in Section IV were used for the experiment. One of the nodes located in a central position was used as a source and two peripheral nodes were used as primary and secondary sinks. The latter was used to receive duplicate copies of critical and reliability-sensitive packets. Every sink was connected to a laptop computer through its USB port. To control the topology on a small surface, we used the power control mechanism provided by the CC2420 driver that enables 31 discreet values

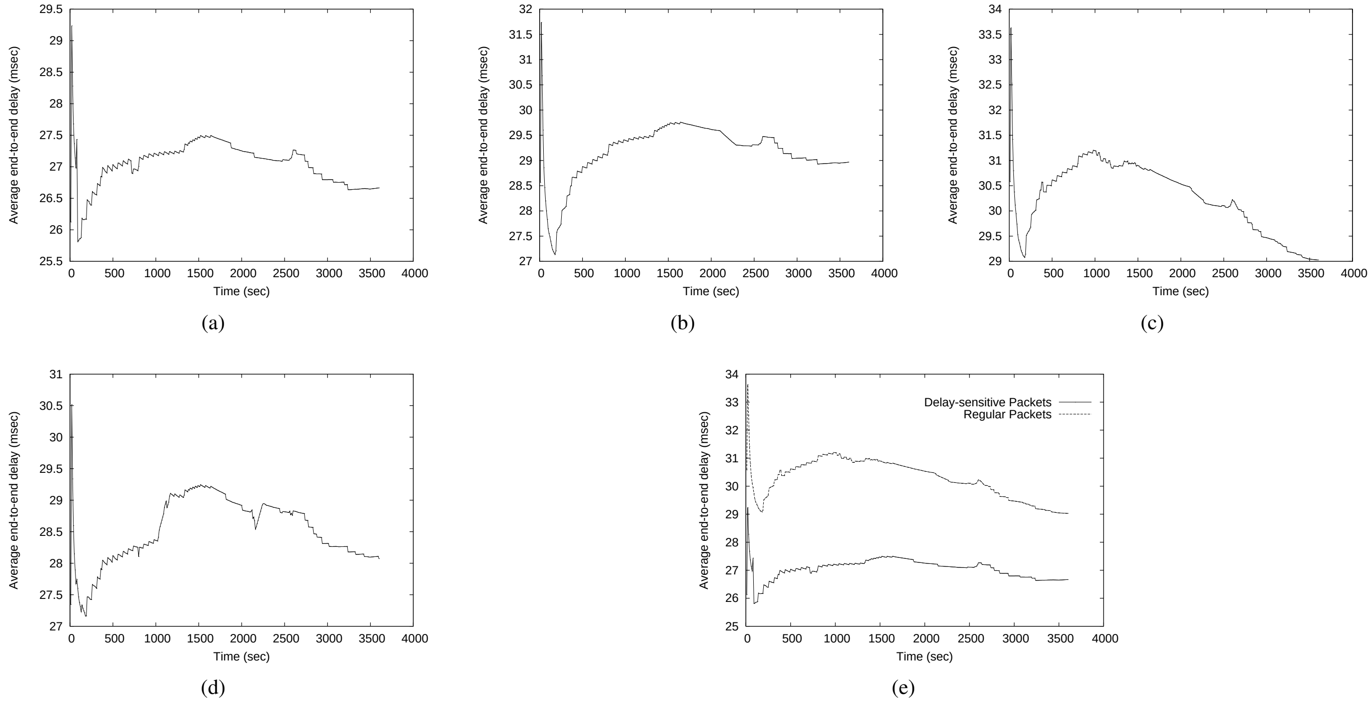

Fig. 2. End-to-end delay: a) delay-sensitive packets, b) critical packet, c) regular packets, d) reliability sensitive packets, e) delay-sensitive vs. regular packets

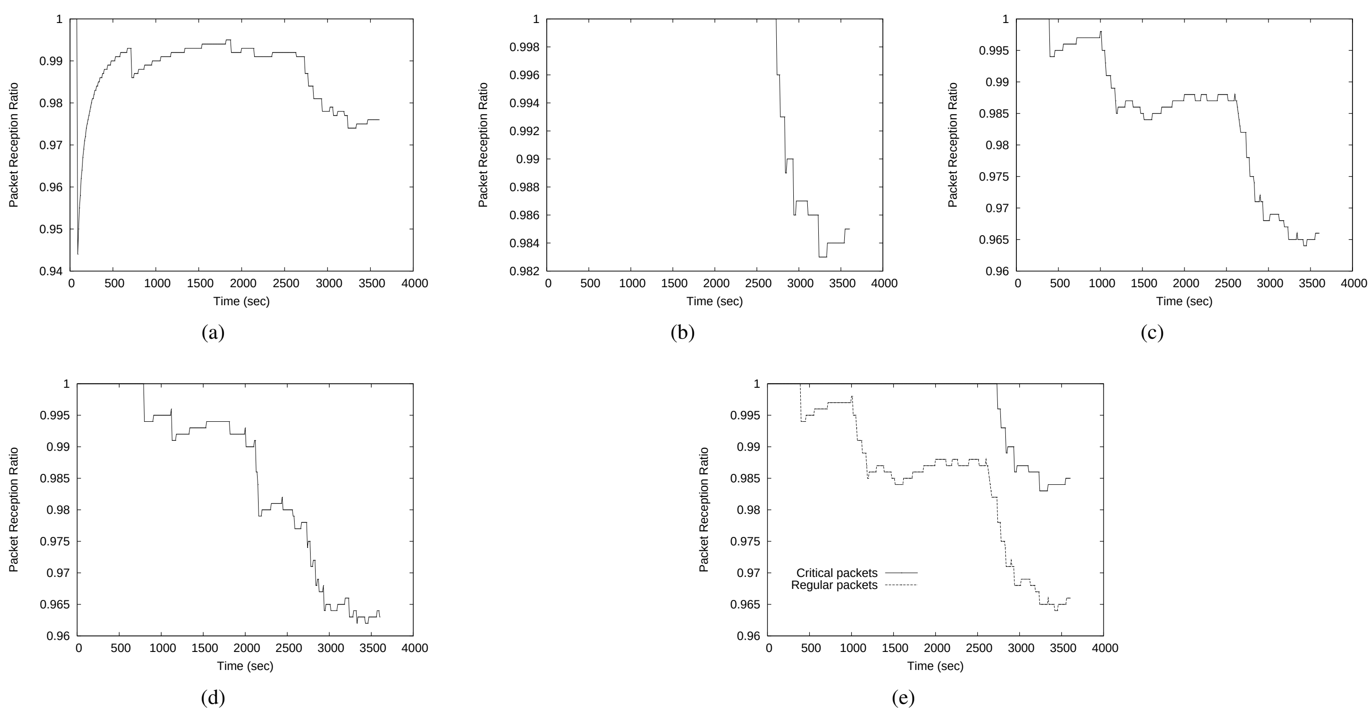

Fig. 3. packet reception ratio: a) reliability-sensitive packets, b) critical packet, c) regular packets, d) delay-sensitive packets, e) critical vs. regular packets

(from 1 to 31). The maximum transmission power was set to 2, resulting in a power range of few tens of centimeters (less than $1\ m$). As shown in Figure 1, the resulting network topology offers an acceptable connectivity and has several multi-hop routes separating the source and the sinks, which helps to comprehensively test the routing protocol. Serial port was used by each sink to construct log files for collecting the simulation results.

The experimental results depicted in Figures 2 and 3 are encouraging and show that a very high rate of packets is always correctly delivered (more than 95%) with reasonable delay (les than 33 ms). More importantly, Figures 2 (e) and 3 (e) show that the protocol provides different QoS for the different packet types, i.e. it ensures higher reliability and lower latency to packets requiring such performances than regular packets. The difference is not very important due to the small size of the network. In more complex scenarios, e.g. with more nodes, routes, etc., the route selection (routing protocol) will have more significant impact on the performance metrics. The protocol is therefore expected to demonstrate even better traffic-based QoS performance differentiation in real applications. This will be checked in next section, where the protocol is evaluated in more complex scenarios, and compares it with state-of-the-art protocols.

VI. Conclusion

A new geographical quality of service (QoS) routing has been proposed in this paper. The protocol makes a traffic-based differentiation and ensures customized QoS according to the packet type, while considering power efficiency. The protocol is suitable for wireless sensor networks with heterogeneous traffic, such as medical and vehicular applications. Unlike most of the routing protocols proposed in the literature, the new protocol has been implemented and evaluated using real sensor motes. TelosB motes were used as a convenient research platform along with Contiki operating system. This operating system has many features and offers simplifications compared to pure event-based systems, which facilitated the implementation of the protocol. A wireless sensor network network of 15 nodes was deployed to measure packet latency and reliability. The results show the protocol provides good QoS and makes the expected traffic differentiation based on QoS requirements.

References

[1] D. Djenouri and I. Balasingham, "Localmor: Localized multi-objective routing for wireless sensor networks," in *The 20th IEEE International Symposium On Personal, Indoor and Mobile Radio Communications (PIMRC'09)*, September 2009, pp. 1188–1192.

[2] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher, "Speed: A stateless protocol for real-time communication in sensor networks," in *IEEE International Conference on Distributed Computing System (ICDCS'03)*, 2003, pp. 46–55.

[3] E. Felemban, C.-G. Lee, and E. Ekici, "Mmspeed: Multipath multi-speed protocol for QoS guarantee of reliability and timeliness in wireless sensor networks," *IEEE Transaction on Mobile Computing*, vol. 5, no. 6, pp. 738–754, 2006.

[4] M. M.-O.-R. Md.Abdur Razzaque, Muhammad Mahbub Alam and C. S. Hong, "Multi-constrained QoS geographic routing for heterogeneous traffic in sensor networks," *IEICE Transactions on Communications*, vol. 91B, no. 8, pp. 2589–2601, 2008.

[5] K. Zeng, K. Ren, W. Lou, and P. J. Moran, "Energy aware efficient geographic routing in lossy wireless sensor networks with environmental energy supply," *Wireless Networks*, vol. 15, no. 1, pp. 39–51, 2009.

[6] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-time power-aware routing in sensor networks," in *In Proceeding of the IEEE International Workshop on Quality of Service (IWQoS)*, 2006.

[7] M. Chen, V. Leung, S. Mao, Y. Xiao, and I. Chlamtac, "Hybrid geographical routing for flexible energy-delay trade-offs," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 9, pp. 4976–4988, 2009.

[8] T. L. Lim and M. Gurusamy, "Energy aware geographical routing and topology control to improve network lifetime in wireless sensor networks," in *IEEE International Conference on Broadband Networks (BROADNETS'05)*, 2005, pp. 829–831.

[9] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," *ACM SIGPLAN Notices*, vol. 35, no. 11, pp. 93–104, 2000.

[10] R. Barr, J. C. Bicket, D. S. Dantas, B. Du, T. W. D. Kim, B. Zhou, and E. G. Sirer, "On the need for system-level support for ad hoc and sensor networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. 2, pp. 1–5, 2002.

[11] L. Girod, J. Elson, T. Stathopoulos, M. Lukac, and D. Estrin, "Emstar: a software environment for developing and deploying wireless sensor networks," in *In Proceedings of the 2004 USENIX Technical Conference*. USENIX Association, 2004, pp. 283–296.

[12] H. Abrach, S. Bhatti, J. Carlson, H. Dai, J. Rose, A. Sheth, B. Shucker, J. Deng, and R. Han, "Mantis: system support for multimodal networks of in-situ sensors," in *WSNA'03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*. New York, NY, USA: ACM, 2003, pp. 50–59.

[13] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *LCN '04: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*. IEEE Computer Society, 2004, pp. 455–462.

[14] D. Djenouri and N. Badache, "An energy efficient routing protocol for mobile ad hoc network," in *The second IFIP Mediterranean Workshop on Ad-Hoc Networks, MedHoc-Nets'03*, Mahdia, Tunisia, June 2003, pp. 113–122.

[15] A. Woo and D. Culler, "Evaluation of efficient link reliability estimators for low-power wireless networks," University of California, Berkeley,, Tech. Rep., 2003.

[16] A. Dunkels, O. Schmidt, T. Voigt, and M. Ali, "Protothreads: simplifying event-driven programming of memory-constrained embedded systems," in *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*. ACM, 2006, pp. 29–42.